WWW.HACKERJOURNAL.IT

HACKER JOURNAL

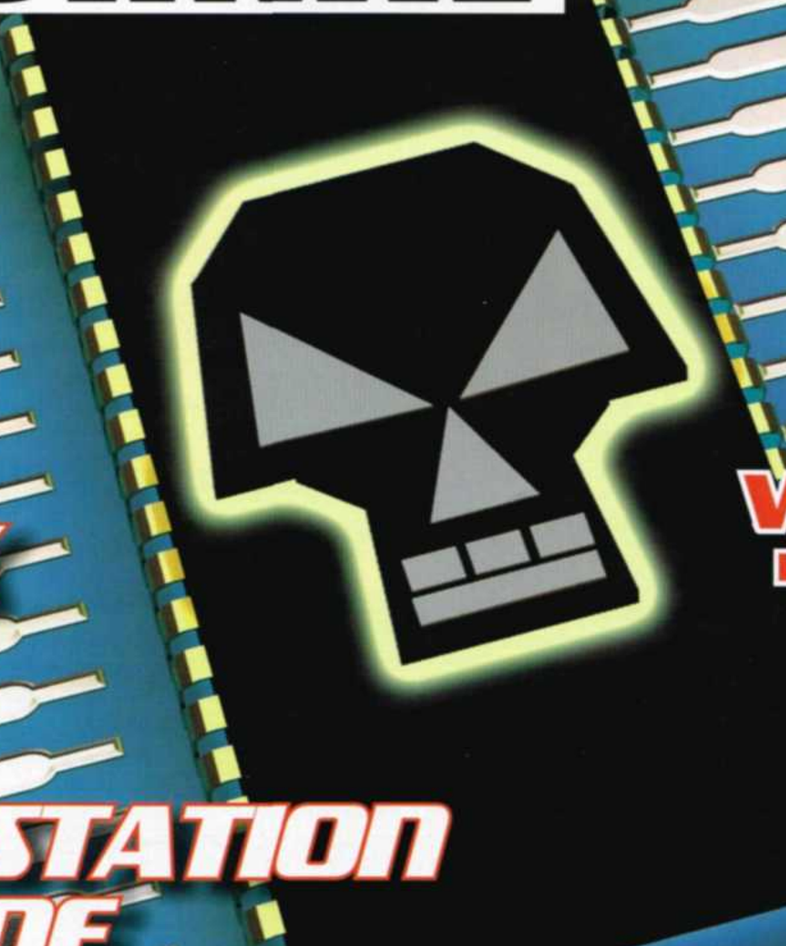
N° 206

DIRECTORY

SESTA PARTE
>CORSO DI C

LABS
>REMOTE SATAN EXE HACK V1.70

CYBERENIGMA
>L'HACKER CRUCIVERBA



IN PRIMO PIANO



VMWARE

IL LATO OSCURO DELLA "MELA"

COME REALIZZARE UN "HACK" SU VMWARE PER CREARE UNA VIRTUAL MACHINE SU MAC

XBOX DS PSP PLAYSTATION IPHONE

TUTTE LE MODIFICHE POSSIBILI

SECURITY LAB

EXPLOIT E DINTORNI

SECURITY LAB

STRATEGIE DI BACKUP

HACKER JOURNAL N° 206 - MENS - ANNO 10 - € 2,00

WLF PUBLISHING

00206

EDITORIALE

FELICI E SCONTENTI

La nuova grafica piace o era meglio la vecchia? Gli articoli più semplici sono graditi o giudicati troppo banali? Sono un po' questi i temi all'indomani della piccola svolta grafica ed editoriale apportata nel numero 205. Partiamo comunque da una premessa doverosa: qualsiasi scelta, in qualsiasi settore, genera consensi ma anche critiche. Accontentare tutti è pressoché impossibile, ma si cerca di trovare una soluzione che, se non accontenta la pluralità dei lettori, quantomeno non li scontenti tutti.
Cerchiamo comunque di fare un po' il punto della situazione in base anche alle vostre mail arrivate copiose in redazione nonostante il periodo estivo. La grafica più "pulita" ed essenziale sembra essere stata gradita. Nessuno ha sentito, almeno per il momento, la mancanza dei teschi che da tempo erano parte integrante dell'iconografia di HJ. Anche il "Laboratorio di sicurezza", per quanto rappresenti un'idea in embrione, ha incontrato il favore di molti lettori, incoraggiandoci a proseguire in questa direzione in parte già tracciata.
Detto questo qualche altro lettore ci ha fatto notare che forse il numero 205 era "troppo semplice". Forse a renderlo meno tecnico del solito ha contribuito non solo la nostra volontà di introdurre articoli più comprensibili al grande pubblico (ampiamente richiesti) quanto l'inconveniente tecnico che ci ha impedito di pubblicare la sesta e ultima parte del corso in C (che peraltro troverete su questo numero).
Se dovessi comunque fare un bilancio tra favorevoli e contrari mi sembra di poter dire che questa piccola revisione di HJ sia stata apprezzata da molti di voi, cari lettori. Se così non fosse vi invito, come sempre, a inviare le vostre critiche e i vostri suggerimenti ai soliti indirizzi di posta.
Fino ad allora, non mi resta che augurarvi buona lettura nella speranza che questo numero 206 sia all'altezza delle aspettative.

Altair

laboratorio@hackerjournal.it
Questo indirizzo è stato creato per inviare articoli, codici, spunti e idee. E' quindi proprio una sorta di "incubatore di idee".

posta@hackerjournal.it
E' l'account creato per l'omonima rubrica che è ricomparsa nelle pagine della rivista. A questo indirizzo dovete inviare tutte le mail che volete vengano pubblicate su HJ.

redazione@hackerjournal.it
Questo è l'indirizzo canonico. Quello con cui potete avere un filo diretto, sempre, con la redazione, per qualsiasi motivo che non rientri nelle due precedenti categorie di posta.

Sommario

Anno 10 - N.206
Ottobre 2010

Editore (sede legale)
WLF Publishing S.r.l.
Socio Unico Medi & Son S.r.l.
via Donatello 71 - 00196 Roma
Fax 063214606

Realizzazione editoriale
Progetti e promozioni Srl
redazione@progettiepromozioni.com

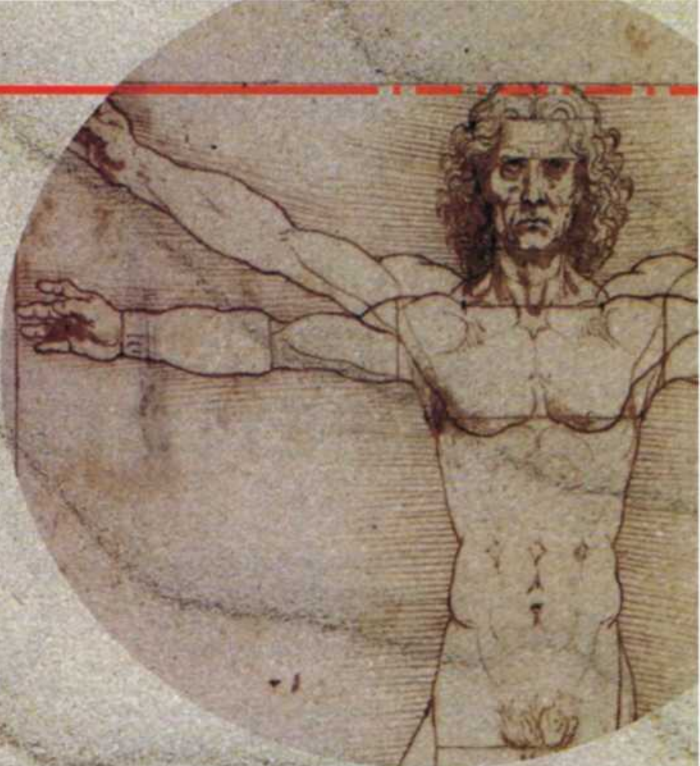
Printing
Grafiche Mazzucchelli S.p.a - Seriate (BG)

Distributore
M-DIS Distributore SPA
Via Cazzaniga 19 - 20132 Milano

Hacker Journal
Pubblicazione registrata al Tribunale di Milano il 27/10/03 con il numero 601.
Una copia: 2,00 euro

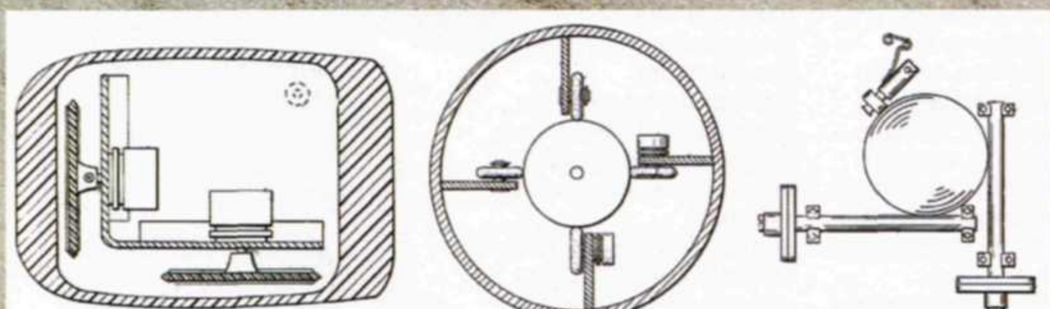
Direttore Responsabile
Teresa Carsaniga
redazione@hackerjournal.it

Direttore Editoriale
Andrea Franchini



Il vecchio
CHE AVANZA

Quando uno pensa alla fantascienza magari è portato a immaginare scenari fantastici come quelli tratteggiati da film cult, come il Pianeta Proibito (il nomignolo scelto dal nostro caporedattore trae spunto proprio da quella pellicola...), oppure da film più "realistici", però lo stesso impregnati di innovazioni tecnologiche, come Blade Runner. Eppure nella realtà le cose non vanno proprio così. Ci troviamo nel 2000 con automobili che non solo non volano, ma utilizzano la stessa combustione dei veicoli introdotti nei primi del '900. Il motore a scoppio si è certo evoluto nel corso degli anni, ma il concetto di base non è poi cambiato così tanto. Chi aspettava il terzo millennio con la speranza di atterrare sul balcone con la propria personale aeromobile dovrà attendere ancora qualche tempo. L'abbiamo presa un po' alla lontana, ma questo nostro excursus ci serviva per fare capire che in tutti i settori l'innovazione progredisce sempre piuttosto lentamente e che spesso c'è la tendenza a rimanere ancorati a vecchie tecnologie che si reputano però affidabili (certo poi ci sono anche ragionamenti di mercato che pesano, ma questo è un altro discorso). Tutto molto sensato. Semmai, la cosa stupefacente è che anche il mondo delle nuove tecnologie, un universo in costante e continua evoluzione, soffra delle stesse tendenze. Capita così che l'interazione uomo/computer avvenga ancora oggi, per la maggior parte, col mouse. Si tratta di un dispositivo introdotto, come concetto, già nel lontano 1967 ed accostato per la prima volta ad un computer nel 1981 quando venne presentato lo Xerox Star, il primo personal computer con il mouse. Sono passati 30 anni eppure il mouse, con qualche piccola evoluzione, è ancora lì. A pensarci bene è una cosa incredibile. Provocazioni a parte, è possibile che in tutto questo tempo non si sia trovata una forma di interazione se non avveniristica, almeno più pratica? Il futuro del mouse forse sarà segnato dalla tecnologia touch screen. L'iPad insegna che probabilmente i computer del futuro non avranno più bisogno del "topo" avendo integrata una tecnologia di comando non solo più innovativa ma anche più intuitiva. Se, come pensiamo, l'avvento dei tablet è destinato a modificare il mondo dei PC consumer, il prossimo passo sarà quello di computer da tavolo interamente touch screen (che peraltro già esistono ma con scarsa diffusione commerciale). Del resto la nuova gamma di iMac, tanto per rimanere in casa Apple, con qualche modifica sarebbe già perfetta per una tecnologia interamente touch di largo consumo. Insomma aspettiamo con ansia il nuovo che avanza e intanto continuiamo a goderci il vecchio che non ci abbandona, e forse va bene così, per cambiare c'è sempre tempo...



Il disegno del primo mouse della storia, derivante dal brevetto di Douglas Engelbart. Era, pensate, il 21 giugno 1967.



ARRESTATO L'HACKER ISERDO

L'FBI ha arrestato un giovane di 23 anni, "Iserdo", autore del kit Butterfly utilizzato dalla botnet Mariposa.

Mariposa è stata sviluppata utilizzando il kit Butterfly, un pacchetto software rivenduto online a un costo tra i 500 e i 1.500 euro, che permetteva a tutti, anche a coloro con ridotte capacità informatiche, di perpetrare il cyber crimine su larga scala. Il kit è stato utilizzato per creare almeno 10.000 esemplari unici di software pericolosi e oltre 700 botnet. Centinaia di istituzioni finanziarie e governative e milioni di aziende private e singoli utenti sono stati attaccati in tutto il mondo.

"Negli ultimi due anni, il software usato per creare la botnet Marisposa è stato venduto a centinaia di criminali, rendendolo uno dei più famosi in tutto il mondo", spiega Robert S. Muller, III, FBI Director. "Queste cyber intrusioni, furti e frodi minano l'integrità di Internet e delle attività che vi si svolgono e minacciano la privacy e il portafogli dei 'naviganti'".

Prezioso il contributo di Defence Intelligence e Panda Security che hanno monitorato il kit Butterfly per circa due anni. "Ciò che colpisce in questo caso, è l'essere giunti direttamente al centro del bersaglio. Infatti, generalmente, gli utilizzatori delle botnet vengono arrestati, ma è estremamente raro raggiungere l'autore che ha sviluppato il kit che sta dietro alla botnet," spiega Christopher Davis, CEO di Defence Intelligence.

"Siamo molto orgogliosi di aver collaborato in questa operazione per sconfiggere cyber criminali come Iserdo, ma sappiamo che si tratta solo di uno dei numerosi hacker che causano danni e il lavoro da svolgere è ancora molto," spiega Juan Santana, CEO di Panda Security.

"Crediamo che la lotta contro il crimine informatico necessiti di uno sforzo collaborativo internazionale da parte dell'industria della sicurezza e delle istituzioni pubbliche. Dobbiamo lavorare per aumentare la conoscenza delle minacce, spingere per una corretta legislazione che preveda dure sanzioni per chi adotta atteggiamenti criminali e garantire una formazione adeguata per i gruppi di lavoro come Mariposa Working Group per sviluppare tecnologie preventive e soluzioni efficaci."

Jeffrey Try, Deputy Assistant Director per l'FBI cyber division commenta "invece di fermare l'uomo che è entrato nelle vostre case, abbiamo arrestato colui che ha fornito la chiave, la mappa e le informazioni sulle migliori abitazioni del quartiere. È un ottimo risultato per le indagini relative al crimine informatico".



ALLA RICERCA DEL "MALIGNO"

Secondo una recente ricerca rilasciata da Barracuda Labs, sulla base di uno studio di due mesi, monitorando più di 25.000 argomenti e 5,5 milioni di risultati di ricerca, Google rimane il più popolare motore di ricerca usato dagli hacker malintenzionati, basandosi sulle parole chiave "avvelenate".
Nel complesso, Google si aggiudica la corona per la distribuzione di malware: più del doppio della quantità di malware rintracciabili su Bing, Twitter e Yahoo!, anch'essi monitorati nella ricerca. Google conquista la vetta con una percentuale del 69%; Seguono: Yahoo! al 18%; Bing al 12% e Twitter solo all'1%.
I primi 10 termini di ricerca utilizzati dai distributori di malware includono il nome di un giocatore NFL, tre attrici e una playmate di Playboy.
"Avvelenare" i risultati dei motori di ricerca è una tattica attiva negli arsenali del crimine informatico da diversi anni. La pratica, conosciuta come blackhat SEO (search engine optimization) è oggi la fonte primaria per dirottare il traffico verso siti pericolosi.



CAMERON DIAZ
È LA CELEBRITÀ PIÙ PERICOLOSA DEL CYBERSPAZIO



Cameron Diaz ha sostituito Jessica Biel come la celebrità più pericolosa da cercare sul Web. Una ricerca di McAfee ha rilevato che la ricerca di immagini più recenti di Cameron Diaz e il relativo download determina un dieci per cento di possibilità di scaricare virus e malware.
Lo studio utilizza i parametri del sito SiteAdvisor, che indica quali siti sono rischiosi per la ricerca di nomi di celebrità sul web e calcola una percentuale globale del rischio. Le prime 10 celebrità di quest'anno con la più alta percentuale di rischio sono:

1	Cameron Diaz
2	Julia Roberts
3	Jessica Biel
4	Gisele Bundchen
5	Brad Pitt
6	Adriana Lima
7	Jennifer Love Hewitt
	Nicole Kidman
8	Tom Cruise
9	Heidi Klum
	Penelope Cruz
10	Anna Paquin

"I cybercriminali seguono gli stessi argomenti più 'caldi' del momento, proprio come i consumatori, e creano trappole in base alle ultime tendenze", ha dichiarato Dave Marcus, security researcher di McAfee Labs. "Se si naviga sul Web da computer o telefono o cliccando sul link da Twitter della propria celebrità favorita, si dovrebbe navigare in modo sicuro e assicurarsi che si stanno utilizzando i software di protezione più recenti."

McAfee®

Problemi DI ARCHIVIAZIONE

Symantec Corp. ha rilasciato i dati relativi all' Information Management Health Check Survey 2010, dai quali emerge che la maggior parte delle imprese non applica le raccomandazioni che esse stesse forniscono quando si parla di gestione dell'informazione. L'87% dei rispondenti crede che sia importante avere un piano formale per la conservazione delle informazioni, ma in realtà solo il 46% ne ha adottato uno. I risultati dell'indagine mostrano che troppe imprese salvano le informazioni a tempo indefinito invece di applicare le policy che permetterebbero loro di cancellare con fiducia i dati o documenti superflui, e pertanto sono affette da un crescente aumento dello storage, tempi eccessivi per il backup di windows, un incremento del rischio di cause giudiziarie e da processi di discovery costosi ed inefficienti.
"Conservazione infinita si traduce in spreco infinito. Le imprese vedono il valore di un solido piano per la gestione delle informazioni, ma in troppi seguono ancora la vecchia abitudine di tenere tutto per sempre" ha dichiarato Brian Dye, vice president of product management, Information Management Group, Symantec. "Il volume dei dati sta crescendo in modo esponenziale, e quindi cercare di conservare tutto causa un grande consumo dello spazio di storage e richiede risorse IT eccessive. Come risultato si spende più denaro e si spreca più tempo per le conseguenze negative di pratiche errate nella gestione delle informazioni e della discovery rispetto a ciò che si spenderebbe cambiando queste pratiche".
Le conseguenze di questi errori nella gestione dell'informazione sono notevoli e prolungate.
I costi per lo storage sono arrivati alle stelle perché l'eccessiva conservazione ha creato una situazione per cui è 1.500 volte più costoso rivedere e risistemare i dati piuttosto che archiviarli, e questo sottolinea ancora una volta l'importanza cruciale per le aziende di avere una buona policy per la cancellazione dei dati e una capacità di ricerca dei dati efficiente.
I backup di windows sono in forte aumento e i tempi per il recovery stanno diventando proibitivi.
Infine considerando la quantità di informazioni archiviate e la difficoltà di accesso alle registrazioni dei backup, l' eDescovery è diventato lento, inefficiente e costoso.



Scoperto il SMS Trojan per smartphone Android

Il primo programma maligno classificato come Trojan-SMS è stato rilevato per gli smartphone con sistema operativo Google Android. Il Trojan-SMS. AndroidOS.FakePlayer.a, questo il nome, ha già infettato un numero di dispositivi mobili.
Il nuovo programma maligno penetra smartphone Android in esecuzione sotto le spoglie di un innocua applicazione/lettore multimediale. Agli utenti viene richiesto di installare un file di poco più di 13 KB con estensione standard di Android .APK. Una volta installato sul telefono, il Trojan utilizza il sistema per iniziare a inviare SMS a numeri a tariffa maggiorata all'insaputa del proprietario o comunque senza il suo consenso, con conseguente passaggio di denaro dal conto di un utente a quello dei criminali informatici.

La categoria Trojan-SMS è attualmente la classe più diffusa di malware per telefoni cellulari, ma Trojan-SMS.AndroidOS.FakePlayer.a è il primo obiettivo specifico la piattaforma Android. Va notato che ci sono già stati casi isolati di dispositivi equipaggiati con Android infettati da spyware. Il primo programma di questo tipo apparve nel 2009.



Giochi pirata sulla PS3: pronto il rilascio del dispositivo

Un gruppo di hacker hardware ha dichiarato di essere in procinto di rilasciare il primo prodotto per consentire ai giocatori di utilizzare giochi pirata su PlayStation 3.
La PS3, nonostante la presenza sul mercato da più di tre anni, ad oggi non è ancora stata violata.
A dire il vero una piccola breccia era stata aperta qualche mese fa da George Hotz, il giovane hacker americano già protagonista del jalbreak dell'iPhone. La sua era una modifica in parte software e in parte hardware piuttosto laboriosa che, tuttavia, non consentiva di utilizzare giochi pirata ma solo di accedere al sistema (trovate tutta la procedura sul numero 196 di HJ). Lo sviluppo di un vero exploit per utilizzare i giochi pirata non è mai seguito.
In questo caso però il gruppo chiamato PSJailbreak ha dichiarato che rilascerà un dongle USB che contiene un software che consente agli utenti di salvare i giochi sul disco rigido della console. Secondo un video degli hacker pubblicato on-line da un distributore australiano, un utente deve solo inserire la chiavetta USB nella console per farla funzionare.
Molti parlano di un bufala, ma c'è chi afferma di avere già testato il dispositivo e di averne accertato il perfetto funzionamento.
Nel caso la notizia fosse reale, c'è poi un risvolto tipicamente legale, infatti la legittimità di prodotti come questo - comunemente chiamati modchips - varia in base al paese.



Massimiliano Brasile
redazione@hackerjournal.it

Labs

LA NOSTRA CASELLA DENOMINATA LABORATORIO@HACKERJOURNAL.IT È UNA FUCINA DI IDEE E PROPOSTE CHE GIUNGONO DAI LETTORI, IN QUESTO BREVE ARTICOLO CONDENSIAMO ALCUNE DELLE PIÙ INTERESSANTI, INVITANDOVI A CONTINUARE AD INVIARE IDEE E MATERIALE.

REMOTE SATANA

Remote satana è un software creato da Simone Cardilli che permette il controllo totale di un computer inviando un file alla vittima con la differenza della totale invisibilità rispetto ad altri software che svolgono le stesse funzioni. Questo software è rivolto a tutti coloro che vogliono controllare il proprio computer da qualsiasi parte del mondo, senza far sapere dall'altra parte che si è osservati. Remote Satana introduce 170 comandi circa tra cui:



Spiare il computer con il suo favoloso keylogger , Aprire siti web, Aprire Software, Visualizzare File, Inviare file , Chat Matrix Chat Windows, Catturare lo schermo, Danneggiare il computer,
Inviare file bat , eseguire comandi bat, Spegnere / Riavviare PC, Schermata FBI e molto altro ancora !

Remote Satana include un Server e il client. Per controllare il computer basta inviare il client (trojan) e sapere l'IP (indirizzo di rete). Con la tecnologia ROBOT non c'è bisogno di selezionare la porta aperta.

Per vedere il software all'opera: http://www.youtube.com/watch?v=pynehMxAylQ

EXE HACK V1.70 FIX

Il programma si può scaricare dal canale di youtube www.youtube.com/user/gicopiro o dal sito www.ldem.it .

Con questo software è possibile:

- ☐ Cambiare icone a programmi già compilati.
- ☐ Unire 2 o più file .exe

□ Unire 2 file qualsiasi con opzioni diverse, come l'aggiunta di un messaggio di errore.
□ Cambiare le info di software.
□ Estrarre icone da programmi compilati.
□ Cambiare una stringa ad un programma (anche per modificare il nome ai suoi controlli).
□ Estrarre altre risorse (come cursori, icone, BMP).

Appena scaricato avremo sul desktop questo file:



Dopo averlo aperto dovremmo cliccare su "vai" e il programma ci darà conferma subito dopo aver creato un altro programma con il nome di "Gico Piro's_ EXE_Hack". Apriamolo e vedremo questa schermata :





Cliccando su "Seleziona l'EXE" potremo scegliere il programma a cui cambiare l'icona, poi procediamo in questo modo:

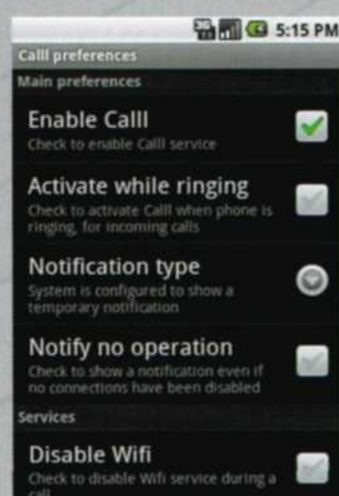
- clicchiamo su "Seleziona l'Icona";
- dopo averla selezionata potremo cliccare su "Cambia!";
- a questo punto dobbiamo solo cambiare il nome all'eseguibile e l'icona sarà modificata.

Dobbiamo rinominare il programma solo se l'utente ha installato sul proprio PC Windows vista o 7. Per cambiare le info di un programma basta cliccare su "Cambia le info di un exe" ., e si aprirà una nuova finestra, clicchiamo su "Apri l EXE" , scegliamo il programma e poi clicchiamo su "Leggi le INFO", così possiamo modificarle. Dopo aver fatto ciò facciamo click su modifica e le info saranno modificate.

CALLL: LO STERMINA RADIAZIONI

Avete idea di quante radiazioni emetta un cellulare? Se proprio volete farvi una cultura in tema visitate il sito http://www.ewg.org/cellphoneradiation/Get-a-Safer-Phone. Qui viene riportata una classifica dei cellulari/smartphone che emettono più radiazioni e che quindi potrebbero essere dannosi per la salute. Il condizionale è d'obbligo visto che non è stato ancora accertato un nesso di casualità certo tra radiazioni e tumori al cervello, certo sapere che il Blackberry 8820 è in testa a questa classifica con emissioni pari a 1,58 watt/chilogrammo non invoglia all'acquisto. Per cercare di arginare il fenomeno delle radiazioni un nostro lettore ha sviluppato un'interessante applicazione per Android che limita i danni portati dalle antenne dei nostri telefoni, semplicemente disabilitando le connessioni di rete (ad ora soltanto Wireless e Bluetooth, in futuro ne sono previste altre) durante una chiamata, per riabilitarle in modo automatico al termine.
L'applicazione si chiama Calll ed è scaricabile gratuitamente dall'Android Market (disponibile sia in inglese che in italiano); maggiori informazioni si possono trovare a questo indirizzo: http://www.cyrket.com/p/android/cambiaso.calll/





Pietro Cinaglia - www.unicz.net/blog
redazione@hackerjournal.it

VMWARE:
IL LATO OSCURO DELLA "MELA"

DISCLAIMER: il presente articolo ha solo scopo informativo; la redazione e l'autore con il rispettivo sito-web citato nell'articolo non si assumono nessuna responsabilità e sconsigliano l'utilizzo della tecnica sotto descritta in quanto, oltre a non essere garantita da Apple, viola il contratto EULA.

MAC

CREARE UNA VIRTUAL MACHINE ALL'INTERNO DI MAC OS X? NON SI POTREBBE MA BASTA UN PICCOLO HACK DI VMWARE...

VMware Workstation e VMware Player sono prodotti Desktop di VMware Inc., compatibili con Windows e GNU/Linux, che consentono di eseguire un sistema operativo in un ambiente virtuale (la "virtual machine", VM) posto all'interno di una macchina reale (definita "host").
Le caratteristiche della VM (numero di processori e cores, disk-size, memoria, ecc...) sono personalizzabili e hanno come limite quello dell'host (se per esempio questo è dotato di 2GB di memoria non si può creare una VM che superi questa dimensione), per questioni legate alle prestazioni, di solito, queste non vengono mai eguagliate.
Nativamente le versioni Workstation e Player (come anche la versione Server e altre...) non sono in grado di eseguire VM con Mac OS X, al contrario della versione Fusion la quale però è compatibile solo con questo; per ovviare a ciò verrà illustrato come effettuare, attraverso una apposita tecnica, un Hack della VM per risolvere il problema di compatibilità.

DIFFERENZE TRA LA VERSIONE WORKSTATION E PLAYER

Senza soffermarci sui particolari possiamo dire che la versione Player (come anche il nome suggerisce) è solo in grado di eseguire VM, mentre la versione Workstation è anche in grado di crearle/modificarle.

REQUISITI DI SISTEMA

- CPU che supporti SSE2-SSE3 e anche la Virtualization Technology: Intel VT o AMD-V
- Windows 2000 o superiore (consigliato XP oppure 7) / Linux (kernel aggiornato)
- 512MB di RAM (consigliati 2GB)
- (minimo) 8GB di spazio libero su disco rigido (consigliati 12GB)

STRUMENTI NECESSARI

- CD/DVD originale Apple Mac OS X (versione consigliata: 10.6 Snow Leopard)
- VMware Workstation 7.0 (free 30-day) e VMware Player (free): http://downloads.vmware.com/
- driver e "darwin.iso" (necessario per il boot): http://www.unicz.net/blog/hackerjournal.php

CREAZIONE DELLA VIRTUAL MACHINE (VM)

Per questa operazione è necessario VMware Workstation che, essendo free per 30 giorni, ci dà tutto il tempo di poter effettuare questa prima parte di creazione della VM (successivamente non sarà più necessario e potremo utilizzare semplicemente la versione Player).

Prima di procedere è suggerito creare un'immagine ISO del CD di Mac OS X in modo da migliorare i tempi di installazione.

(Nota: per passare da un punto all'altro è sempre richiesto il click sul pulsante 'Next')

1. Avviamo il software e clicchiamo su 'File' => 'New' => 'Virtual Machine...' per iniziare la creazione della nostra macchina virtuale;
2. nella prima schermata scegliamo la modalità 'Custom (advanced)';
3. impostiamo come 'Hardware Compatibility:' quella di default ('workstation 6.5-7.0') in modo da avere tutte le funzionalità della nuova versione.
4. scegliamo di installare in un secondo momento il sistema operativo ('I will install the operating system later.');
5. non essendo Mac OS X nativamente riconosciuto da VMware Workstation (player e server) selezioniamo '6.Other' e come versione 'Other';
6. diamo un nome alla nostra VM (per esempio "Mac") e, se non necessario, lasciamo invariata la 'Location' (di default è una cartella 'vmware' posta nel disco in cui è stato installato l'omonimo software);
7. particolarmente importante (anche se può essere successiavamente modificata) è la scelta del numero di processori e cores per ognuno di questi da utilizzare, l'impostazione di default va già bene...;
8. la quantità di 'Memory' (memoria RAM) da dedicare alla VM dovrebbe essere di almeno 512 MB anche se il consiglio è di arrivare a 1GB (1024 MB), in ogni caso è consigliato non superare il 50% della risorsa massima disponibile in modo da non rallentare eccessivamente il sistema Host;
9. per quanto riguarda la connessione di rete è consigliato scegliere l'impostazione 'Use bridge Networking';
10. *in* 'I/O Adapter Types' selezioniamo 'LSI Logic';
11. scegliamo di creare un nuovo disco virtuale ('Create a new virtual disk') impostando, nella successiva schermata, 'Virtual Disk Type' a 'SCSI' e lasciando invariato 'Mode';
12. la dimensione del disco è una caratteristica che non può essere facilmente modificata in seguito perciò è opportuno avere già le idee ben chiare su quanto spazio si voglia dedicare alla VM, ricordando che si ha comunque la possibilità di condividere file tra Host e Guest e, se necessario, montare una partizione all'interno di quest'ultimo. Consiglio come spazio almeno 12GB e di scegliere 'Split virtual disk into 2GB files' per facilitare un eventuale spostamento (è importante notare che non tutti gli XGB scelti verranno subito allocati ma questa quantità verrà considerata come un limite da non oltrepassare);
13. lasciamo invariata la schermata 'Disk file' e procediamo visualizzando il riepilogo delle caratteristiche impostate;
14. clicchiamo su 'Customize Hardware...':
15. rimuoviamo dall'elenco 'Floppy'
16. attraverso il tasto 'Add...' aggiungiamo un 'USB Controller' (le impostazioni di default vanno bene)
17. selezioniamo dall'elenco su 'New CD/DVD (IDE)' e scegliamo come 'ISO Image' il file "darwin.iso" che avrete in precedenza scaricato;
18. Completiamo l'operazione con un click sul tasto 'Finish'

Nel percorso (Location) scelto al punto 6 è presente una cartella con il nome della nostra VM contenente diversi file, tra cui uno con estensione ".vmx" che dovrà essere opportunamente modificato; una volta aperto (può essere editato con "blocco notes" in windows oppure "gedit" in linux) copiamo queste stringhe al suo interno:

```
ethernet0.virtualDev =
"e1000"
ich7m.present = "TRUE"
smc.present = "TRUE"
tools.syncTime = "TRUE"
ft.secondary0.enabled =
"TRUE"
tools.upgrade.policy =
"upgradeAtPowerCycle"
usb:0.present = "TRUE"
usb:1.present = "TRUE"
usb:1.deviceType = "hub"
usb:0.deviceType = "mouse"
keyboardAndMouseProfile =
"macProfile"
```

e sostituiamo

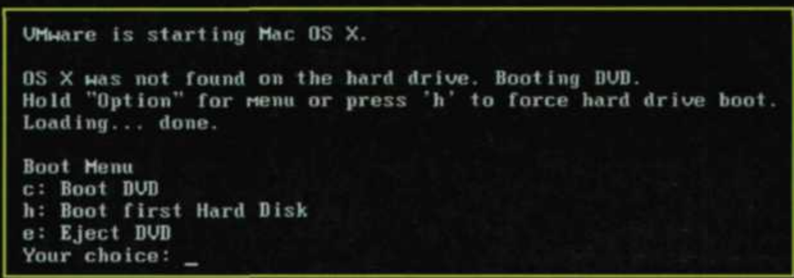
```
guestOS = "other" =>
guestOS = "darwin10"
```

Salviamo e chiudiamo il file.

INSTALLAZIONE SISTEMA OPERATIVO

Avviamo finalmente la nostra VM, pronta ad ospitare il sistema operativo della mela, cliccando su Power on this virtual machine. Durante l'avvio premere continuamente il tasto [INVIO] fino alla visualizzazione del Boot Menu (se dovessero comparire errori riavviare e ripetere l'operazione). Cliccando la combinazione di tasti Ctrl+Alt otterremo nuovamente la disponibilità del puntatore mouse sulla macchina Host, possiamo ora cambiare l'immagine ISO corrente con quella del sistema operativo preparata in precedenza; per cambiare ISO Image cliccare sull'icona del cd (in basso alla schermata) con il tasto DX del





mouse e selezionare Settings... una volta eseguito il cambio, clicchiamo nuovamente sulla schermata di VMware per rientrare nel sistema Guest e selezioniamo Boot DVD inserendo come valore "c" [INVIO]; verrà avviato Mac OS X.
Nella prima schermata ci verrà chiesto di selezionare la lingua e proseguire.
Ora dobbiamo creare una partizione nel nostro Virtual Disk, utilizzeremo perciò i tool messi a disposizione da Apple; perciò selezioniamo dal menu in alto Utility => Utility Disco...
Selezioniamo dal menu a sinistra il nostro unico disco e poi attraverso il menu Inizializza diamo un nome alla nuova partizione (es. "MAC") e clicchiamo su Inizializza, per quanto riguarda il formato lasciamo il valore Mac OS esteso (Journaled).
Possiamo procedere ora con l'installazione guidata...
Terminata l'installazione consigliamo la creazione di una Snapshot per salvare lo stato del sistema.

CREAZIONE/ RIPRISTINO DI UNA SNAPSHOT ("ISTANTANEA")

Le snapshot sono uno strumento particolarmente utile quando si effettuano modifiche al sistema (come installazione di nuovi driver o aggiornamenti) in quanto salvano lo stato attuale permettendo di ripristinarlo in seguito; è come se venisse fatta una fotografia al sistema e conservata...

E' importante ricordare, però, che una volta ripristinata una Snapshot non si può tornare allo stato precedente, a meno che non sia presente una Snapshot anche per questo.



- La creazione è estremamente semplice, basta cliccare su VM => Snapshot => Take Snapshot... per accedere al menu di creazione (è possibile, opzionalmente, inserire anche una descrizione dello stato salvato e modificarne il nome).

- Il ripristino avviene sempre attraverso VM => Snapshot ma scegliendo Revert to Snapshot e selezionando dall'elenco l'istantanea desiderata.

FILE ".PKG"

i file pkg sono pacchetti di installazione singoli, per avviarli è sufficiente effettuare un doppio click sull'icona del file interessato e verrà avviata l'installazione grafica.

DRIVER VIDEO E ABILITAZIONE DELL'ACCELERAZIONE 3D

I driver Vmsvga2 possono essere considerati una versione patchata dei Driver SVGA II installabili con i VMware Tools. Grazie ad essi è possibile modificare la risoluzione dello schermo, abilitare l'accelerazione 3D ed aumentare la fluidità con cui i video e le immagini vengono visualizzate nella VM.
Dopo l'installazione dei driver è richiesto il riavvio del sistema al termine del quale possiamo modificare la risoluzione dello schermo direttamente da Preferenze di sistema, collocato nella barra di Mac OS X, => Monitor.
Icona "Preferenze di sistema":



Per abilitare l'accelerazione 3D spegniamo la VM e da "Edit Virtual machine Settings", clicchiamo sulla voce "Display" e spuntiamo l'opzione "Accelerate 3D Graphics"; clicchiamo su "Save" per salvare le modifiche effettuate.

DRIVER MOUSE

Procediamo quindi ad estrarre l'archivio riguardante i driver del mouse e copiare ApplePS2Controller.kext in system/library/extension (potrebbero comparire errori riguardo l'errata installazione, infatti ancora bisogna finire...).
Eseguire da terminale (nella barra: Applicazioni => Utility => Terminale) i seguenti comandi (sarà richiesta la password di amministratore):
•sudo chmod -R 755 / System/Library/Extensions/ ApplePS2Controller.kext
•sudo chown -R root:wheel / System/Library/Extensions/ ApplePS2Controller.kext
Al termine installare "VMMouse_ v1.2.1_Common_Installer.pkg" (Durante l'installazione dei driver mouse assicurarsi che il valore di default sia impostato su "VMmouse for ApplePS2Controller")

INSTALLAZIONE VMWARE TOOLS

Il pacchetto dei VMware Tools è contenuto all'interno dell'ISO-Image "darwin.iso".
Affinché sia possibile eseguirne l'installazione è necessario riconnettere, una volta avviato Mac OS X, il device CD/DVD (VM => Removable Devices => CD/DVD (IDE) => Connect) e attendere che sul Desktop compaia l'icona "VMware Tools";
cliccando su di essa verrà aperta una finestra di dialogo da cui sarà possibile avviare l'installazione attraverso l'icona "Installa VMware Tools".

ALTRI DRIVER/ PACCHETTI

Possono essere installati semplicemente con i rispettivi file ".pkg".

CONDIVISIONE FILES E CARTELLE

La condivisione di files e cartelle può avvenire attraverso la rete esistente tra macchina Host e Guest, perciò inserendo l'indirizzo locale dell'Host (del tipo 192.168.1.4) in "Vai" => "Connessione al server..." è possibile abilitare la condivisione (sempre che questa sia stata configurata anche nell'Host; dipende dal sistema operativo).

SHRINK DEI FILE ".VMDK"

I file contenenti l'hard disk virtuale (con estensione ".vmdk") aumentano la loro dimensione, fino al limite scelto in fase di creazione, ogni volta che vengono creati/inseriti nuovi file all'interno della VM. Al contrario durante l'eliminazione dei file, la loro dimensione non diminuirà e per recuperare spazio bisogna ricorrere a una operazione di Shrink ("riduzione").
Per fare ciò:
eseguite, attraverso il terminale, il seguente comando e attendete il completamento:

```
• dd if=/dev/zero of=/zerofile bs=4096k count=1000000; sync; sleep 1; sync; rm -f / zerofile
```

Se avete installato i VMware Tools utilizzate, sempre da terminale, eseguite il comando:

```
• /Library/Application\ Support/VMware\ Tools/
vmware-tools-guestd -cmd "disk.shrink"
```

altrimenti spegnete la VM e da Edit Virtual Machine Settings => Hard Disk (SCSI) eseguite la funzione Utilities => Compact

Entrambe le operazioni (con o senza VMware Tools) potrebbero durare diversi minuti...

ERRORI/PROBLEMI COMUNI

• Durante il riavvio può essere visualizzato un errore riferito alla CPU; riavviare il sistema da VMware.
• Tornando nella VM dopo aver lavorato per diverso tempo nell'Host o in seguito all'avvio dello screensaver, mouse e tastiere potrebbero non funzionare; riavviare il sistema da VMware; per risolvere quasi totalmente tutti i problemi dovuti a mouse e tastiere consiglio di installare gli appositi driver (i malfunzionamenti sono dovuti al fatto che non si usa una tastiera compatibile con i driver preinstallati).
• Se Mac OS X non si connette a internet: spegnere la VM e spuntare, tra le impostazioni del "Network Adapter", la voce "Replicate physical network connection state".
• Se avviando la VM vengono comunicati errori riguardanti i driver audio, disattivare la scheda audio virtuale sino a quando non verranno installati i driver in Mac OS X (per la disattivazione: Edit Virtual Machine Settings => Sound card disabilitare "Connected" e "Connected at power on").

File a supporto dell'articolo
La documentazione di supporto all'articolo la trovate all'indirizzo http:// www.unicz.net/blog/ hackerjournal.php

di Spirit
redazione@hackerjournal.it

STRATEGIE DI BACKUP

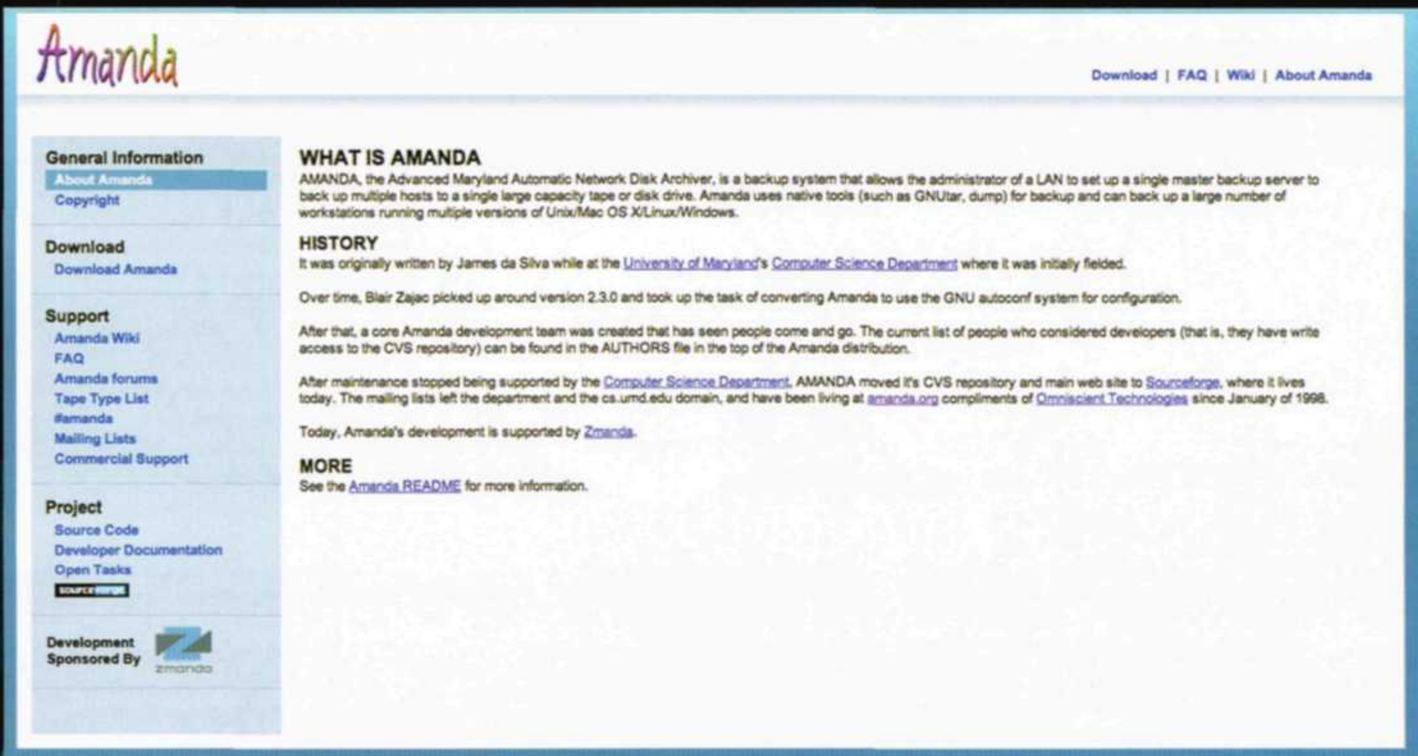
Nell'utilizzo professionale di una soluzione software, uno dei requisiti da considerare in fase di progettazione è un adeguato sistema di backup dei dati, indispensabile per ripristinare le condizioni originali della macchina in caso di problemi. Il problema del backup non deve essere sottostimato nemmeno dagli utenti casalinghi, per i quali il rischio di perdere dati è un problema non banale, che spesso porta con sé spreco di tempo per rifare lavori già compiuti in passato e persi per svariati motivi.
Le soluzioni di backup sono molteplici e non possono essere trattate in maniera completa in un solo articolo. In ogni caso, tutte comportano scelte simili ed hanno in comune diversi fattori, perciò lo scopo principale di quest'articolo è di mostrare le casistiche più comuni ed insegnare a prendere una decisione adatta alla propria situazione.
Il trucco più semplice per trovare una giusta soluzione di backup è "pensare sempre alla peggiore situazione" (ndr: la legge di Murphy): salvare i dati nel proprio disco fisso non garantisce alcuna sicurezza, e tutti coloro che lavorano con un sistema informatico devono sempre adottare una politica di backup adeguata al valore del contenuto del proprio lavoro.
Spesso si tende a sottovalutare proprio quest'ultimo fattore, ma è fondamentale restituire l'adeguato valore al tempo impiegato per produrre un qualsiasi documento o dato. Questo scrupolo deve valere per tutti, dagli utenti casalinghi, alle grandi aziende e una strategia di backup deve essere studiata sin dalle prime fasi di lavoro.
Linux, data la sua natura di sistema adatto come server, ci viene incontro con una serie di utility che in parte derivano dal patrimonio Unix. Il software indiscutibilmente più utilizzato per i backup e l'archiviazione dei file è il comando tar, che in questo articolo utilizzeremo nella sua versione GNU per Linux, disponibile per ogni distribuzione.
Il primo passo per creare un backup del contenuto di una directory è creare un archivio mantenendo i permessi e le caratteristiche originali dei file da salvare. Per creare un archivio compresso con il metodo bzip2 basterà eseguire da shell testuale il seguente comando:

```
# tar cvfjp /backup-`date
+%d-%m-%y`.tar.bz2 /etc
```

Questo semplice comando prenderà tutto il contenuto della directory /etc e lo inserirà nel file compresso backup-[datacorrente].tar.bz2, mantenendone i permessi originali. Per essere sicuri di un qualsiasi sistema di backup, occorre testare accuratamente

DATI

"Le possibilità che i dati vadano perduti sono direttamente proporzionali alla loro importanza".



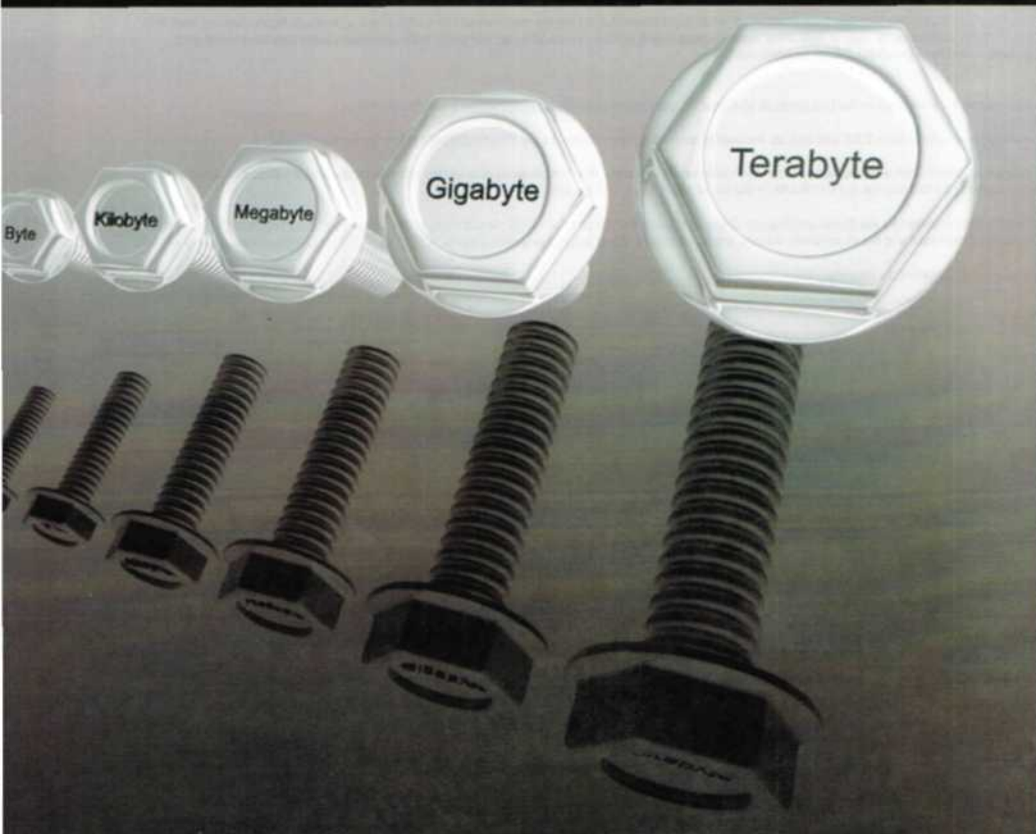
se il metodo permette il ripristino del contenuto voluto. Per decomprimere il nostro archivio lo spostiamo in una directory temporanea ed eseguiamo ancora il comando tar:

```
# mv /backup-[data].tar.bz2 /tmp
# cd /tmp
# tar xvfjp backup-[data].tar.bz2
```

Se tutto funziona correttamente, allora possiamo fidarci del metodo, altrimenti analizzeremo l'errore per trovare una soluzione. Solo testando un backup ed il relativo ripristino possiamo tutelarci anche in caso di reale bisogno, quindi questo passaggio, spesso considerato inutile, va effettuato ogni volta che è introdotto qualcosa di nuovo nel metodo. Un backup come quello mostrato precedentemente servirà in caso di cancellazione errata di uno o più dati all'interno del filesystem. Questo problema capita spesso, ma non è l'unico nemico dal quale dobbiamo difenderci.
Nonostante le tecnologie costruttive dei supporti permanenti (es. dischi fissi, cd-rom) siano migliorate parecchio nel tempo, non si potrà mai avere un'affidabilità assoluta.
Per questo motivo, un backup serio non deve essere mai salvato sullo stesso supporto e nella stessa macchina dei dati destinata a preservare potenziali guasti: in caso di problemi "meccanici", perderemo i dati originali e l'archivio di backup rendendo inutile tutti gli sforzi compiuti. Per un utente casalingo, può bastare un salvataggio periodico dei propri dati su DVD, che si prestano bene allo scopo dato il basso costo dei supporti e di un buon masterizzatore.
Prima di procedere dovremo decidere cosa salvare del nostro disco fisso, scegliendo tra un salvataggio di tutto il sistema o dei soli documenti importanti degli utenti. Data la struttura del sistema operativo Linux, che permette di salvare tutto il contenuto del disco e di ripristinare tutto come era in origine semplicemente copiando i file, consiglio di fare almeno una volta un backup completo. Quando abbiamo a disposizione un sistema perfettamente configurato e funzionale alle nostre esigenze, possiamo evitare con un semplice backup la procedura di installazione da zero tipica di un qualsiasi sistema Windows dopo formattazione del disco.
Per fare un backup di tutto il sistema, ancora con tar, basterà inserire nell'archivio tutte le directory tranne quella chiamata /proc. Per ripristinare la situazione originale basterà decomprimere tutti i file inseriti nel backup precedente, creare da zero la directory /proc con i permessi 555, utente e gruppo root. Dopo aver recuperato i file dovremo ripristinare il boot loader e potremo quindi riavviare la nostra macchina come se il disastro non fosse mai avvenuto. Questa procedura non è fondamentale solo in caso di ripristino, ma è utile anche quando si devono predisporre un grosso numero di macchine con hardware identico: possiamo installare Linux su una di queste ultime e fare la copia del sistema in tutte le altre, riducendo considerevolmente i tempi di configurazione di ogni singola macchina.

GRANDE AZIENDA, GRANDE ARCHIVIO

Sebbene questa soluzione sia abbastanza flessibile, in caso di backup eseguiti in



una LAN aziendale non si dimostra abbastanza pratica. I masterizzatori sono supporti economici, ma non coprono grandi capacità e per questo motivo in ambito professionale sono sostituiti dai nastri magnetici. I nastri hanno capacità molto elevate, a discapito della velocità di lettura/scrittura, operazioni che avvengono in maniera sequenziale.
Riguardo la questione hardware, esistono diverse soluzioni con differenti fasce di prezzo, in base al numero di funzioni fornite ed al numero di nastri contenuti contemporaneamente. Per quanto riguarda i DAT, esistono diversi standard chiamati DDSn, dove n è un numero incrementale che contraddistingue la versione dello stardard. Le unità compatibili con uno standard DDSn supportano anche le versioni precedenti (x < n), ma non è vero il contrario. Rispolverando ancora una volta il tar, nato inizialmente proprio per facilitare il backup su nastro, basterà eseguire questo comando per fare il backup di una directory o di tutto il sistema:

```
# tar cvf /dev/st0 /
directory
```

Dove /dev/st0 è il dispositivo relativo al drive per i nastri. Per quanto riguarda il parco software per i backup, oltre al tar ed al già citato mkcdrec, esistono soluzioni commerciali molto famose portate su Linux, come ad esempio BRU e i prodotti della Veritas, ma anche soluzioni libere come Amanda, dalle enormi potenzialità.
In questo articolo mostreremo l'uso di Amanda data la sua praticità e disponibilità. Questo software si presenta come una soluzione per il backup che permette agli amministratori di una LAN di creare un server master per il backup, per salvare i contenuti dei dischi di client multipli, in un solo drive a nastro ad alta capacità. Con le nuove versioni, inoltre, è possibile usare Samba per fare il backup di client e server.
Il sito del progetto è www.amanda.org, dove è possibile scaricare il software in formato sorgente. Il programma funziona creando uno spazio sul disco del backup server chiamato "holding disk", dove i client scrivono i loro backup in parallelo per ottimizzare le prestazioni. Un processo indipendente procede poi a sistemare il contenuto del disco holding nel nastro vero e proprio. Per utilizzare Amanda possiamo installare direttamente i pacchetti per la nostra distribuzione oppure compilare i sorgenti scaricandoli dal sito. Per Debian, l'installazione può avvenire come segue:

```
# apt-get install amanda-
client
```

Mentre per il server:

```
# apt-get install amanda-
server
```

Per i client, il processo di installazione colloca automaticamente una riga all'interno del file /etc/inetd.conf, per inserire il demone che invierà i file al backup server:

```
amanda dgram udp wait
Amanda /PATH/libexec/
amandad amandad
```

Se si installa dai sorgenti, quindi, bisogna aggiungere a mano questa riga. Amanda permette di fare backup completi ed incrementali seguendo un ciclo apposito: l'amministratore dovrà quindi decidere quanti backup

incrementali fare intercorrere tra un backup completo ed un altro. La decisione non è banale, perché eseguire spesso backup completi comporta maggiore necessità di spazio su nastro (da ricordare che Amanda può archiviare dati anche su disco, la scelta spetta all'amministratore di sistema), ma velocizza il processo di ripristino e la sicurezza della sua riuscita.
I backup incrementali funzionano salvando solo le modifiche apportate al sistema dal backup precedente, quindi sono molto più veloci ed occupano meno spazio. In fase di configurazione dobbiamo quindi stabilire il "ciclo di dump", che rappresenta il numero massimo di backup incrementali che intercorre tra due backup completi. Il software può decidere di eseguire prima del termine massimo un backup completo, se le condizioni lo rendono opportuno.

A questo punto procediamo con la configurazione di base di Amanda: il file che contiene le direttive principali si chiama amanda.conf e si trova in una sottodirectory che identifica il set di configurazioni, per permettere l'utilizzo di impostazioni aggiuntive differenti per diversi tipi di backup della stessa azienda. Sotto Debian il set di base si chiama "DailySet1", e possiamo editare il file /etc/amanda/DailySet1/amanda.conf modificandolo per aderire alle nostre esigenze.

Ogni nuovo nastro deve essere inizializzato tramite il comando "amlabel", per far sapere ad Amanda che può utilizzarlo:

```
# amlabel Daily Daily-123
slot 123
```

A questo punto, per ogni host, dobbiamo decidere di quali dispositivi fare il backup. Modifichiamo quindi il file /etc/amanda/DailySet1/disklist, inserendo una voce per ogni dispositivo con la seguente sintassi:

```
hostname diskname [
diskdevice ] dumptype [
spindle [ interface ] ]
```

Diskname è un nome identificativo del disco da salvare, mentre diskdevice può essere un dispositivo o un mount point. In questo modo è possibile salvare solo i dati di un certo applicativo, come ad esempio un gestionale. Un esempio di voce da inserire nel file disklist può essere:

```
host1    /var/lib/postgres/
comp-high
```

Nel caso di un'azienda con software eseguiti sotto operativi Windows si deve usare Samba, e nella riga del file disklist corrispondente al client si deve inserire, al posto dell' host, una macchina linux sulla quale Samba è installato, e al posto del diskdevice si deve inserire il path corrispondente al filesystem condiviso, ad esempio //nomepcwin/C$. In questo caso bisogna condividere C:\ nel client Windows, affinché Amanda possa accedervi attraverso Samba, che può essere installato anche sullo stesso backup server. Una volta terminata la configurazione, bisogna inserire una voce nel file /etc/crontab per eseguire automaticamente il comando amdump, che si preoccupa di fare il dump di tutte le voci inserite nel file disklist, ad esempio:

```
45 0 * * 2-6    /usr/sbin/
amdump DailySet1
```

Questa operazione automatica è possibile solo nel caso si possieda un dispositivo di nastri che carica automaticamente un nuovo nastro al termine di quello precedente, in caso contrario bisogna stabilire una politica aziendale che deleghi al personale il compito di cambiare nastro in maniera periodica, in base alla capacità ed al volume complessivo del backup. Queste operazioni manuali sono in genere da sconsigliare, perché necessitano di un controllo attento e fanno aumentare il rischio di non procedere con il backup per una dimenticanza.
Come al solito dopo il primo backup con un nuovo strumento è necessario provare il recupero dei dati, che con Amanda si fa tramite l'utility amrecover. Una volta eseguito questo comando, avremo a disposizione un'interfaccia a linea di comando che permette di navigare tra le directory e di recuperare anche parzialmente i dati registrati su nastro. Tramite il comando "add" si aggiungono i file e le directory da inserire nella lista dei file da recuperare, e tramite "extract" si esegue il ripristino, durante il quale sarà richiesto il nastro che contiene i dati cercati.

CONCLUSIONI

Tramite questa soluzione è possibile installare su un server Linux di moderate risorse hardware un sistema di backup professionale, senza costi di licenza per il sistema operativo e per il software di gestione dei nastri, in media molto elevati.
Termina qui il nostro breve viaggio verso alcuni dei sistemi di backup utilizzabili sotto Linux.
L'ultimo consiglio che posso darvi è di tenere sempre in considerazione quanto già detto in precedenza: un backup deve fornire la sicurezza di poter avere a disposizione in ogni momento il lavoro già svolto, ed ogni utente abbastanza furbo dovrebbe provvedere almeno ad una soluzione minima.
Per questo motivo, oltre a salvare i dati, è fondamentale farlo spesso, come consiglia la simpatica Legge di Pickering sui dati perduti: "La probabilità che i dati sul disco rigido si cancellino è direttamente proporzionale al tempo trascorso dall'ultimo backup".

di Spirit
redazione@hackerjournal.it

EXPLOIT E DINTORNI

Questa nuova sezione di Security Lab (il nostro inserto dedicato in modo mirato ai temi della sicurezza, lanciato a partire da questo numero) è dedicata ai malware, exploit e attacchi che si sono guadagnati l'onore delle cronache nell'ultimo mese.

Malware individuati nei computer degli utenti

Posizione	Nome	Computer infettati
1	Net-Worm.Win32.Kido.ir	261718
2	Virus.Win32.Sality.aa	174504
3	Net-Worm.Win32.Kido.ih	158735
4	Net-Worm.Win32.Kido.iq	119114
5	Exploit.JS.Agent.bab	108936
6	Trojan.JS.Agent.bhr	104420
7	Worm.Win32.FlyStudio.cu	80196
8	Virus.Win32.Virut.ce	59988
9	Trojan-Downloader.Win32.VB.eql	47798
10	Worm.Win32.Mabezat.b	40859

Posizione	Nome	Computer infettati
11	Trojan-Dropper.Win32.Flystud.yo	31707
12	Worm.Win32.Autoit.xl	31215
13	P2P-Worm.Win32.Palevo.aomy	30775
14	P2P- Worm.Win32.Palevo.fuc	26027
15	Exploit.JS.CVE-2010-0806.aa	25928
16	P2P-Worm.Win32.Palevo.aoom	25300
17	Hoax.Win32.ArchSMS.ih	24578
18	Trojan.Win32.AutoRun.ke	24185
19	Packed.Win32.Katusha.n	23030
20	Trojan-Downloader.Win32.Geral.cnh	22947

Malware diffusi via Internet

Posizione	Nome	Computer infettati
1	Exploit.JS.Agent.bab	169086
2	1 Trojan-Downloader.JS.Pegel.bp 74504	123446
3	Exploit.Java.CVE-2010-0886.a	65794
4	AdWare.Win32.FunWeb.q	58848
5	Trojan-Downloader.VBS.Agent.zs	58591
6	Trojan.JS.Agent.bhr	57978
7	Exploit.Java.Agent.f	53677
8	Trojan-Downloader.Java.Agent.fl	53468
9	AdWare.Win32.FunWeb.ds	45362
10	Trojan.JS.Agent.bhl	45139

Posizione	Nome	Computer infettati
11	AdWare.Win32.Shopper.l	37790
12	Exploit.HTML.CVE-2010-1885.a	36485
13	AdWare.Win32.Boran.z	28852
14	Exploit.Win32.IMG-TIF.b	28238
15	Exploit.JS.Pdfka.bys	28084
16	Trojan.JS.Agent.bmh	27706
17	Exploit.JS.CVE-2010-0806.aa	26896
18	Exploit.JS.Pdfka.cny	26231
19	AdWare.Win32.FunWeb.ci	26014
20	Trojan.JS.Redirector.cq	26001

Worm.Win32.Autoit.xl

Molto attivo nell'ultimo periodo il malware Worm.Win32.Autoit.xl, è in sostanza costituito da uno script maligno elaborato in linguaggio AutoIt, preposto all'esecuzione di numerosi task dannosi per i computer degli utenti: disattivazione del firewall di Windows, applicazione di regole inibitorie, download ed installazione di ulteriori programmi maligni. E' interessante osservare come quasi un quarto dei casi di rilevamento e neutralizzazione di tale malware si sia prodotto in Brasile. Circa la metà dei rilevamenti, invece, ha avuto luogo in Russia ed in Ucraina.

La nuova variante «aa» di Exploit.JS.CVE-2010-0806

E' un exploit in grado di sfruttare la vulnerabilità CVE-2010-0806, individuata nel mese di marzo dell'anno in corso. I malintenzionati fanno attualmente sempre più ricorso all'applicazione di processi di offuscamento degli script, così come a metodiche di antiemulazione; ciò genera, ovviamente, la progressiva comparsa di nuove varianti del suddetto exploit. Ricordiamo, con l'occasione, come la vulnerabilità CVE-2010-0806 sia altresì utilizzata da due ulteriori programmi maligni: Exploit.JS.Agent.bab e Trojan.JS.Agent.bhr.

Hoax.Win32.ArchSMS.ih

Questo singolare programma maligno si è reso protagonista dell'introduzione di un metodo del tutto nuovo al fine di ingannare gli utenti della Rete. In genere il programma viene distribuito camuffato sotto forma di software gratuito e apparentemente legittimo. Al momento della sua apertura compare sullo schermo una finestra nella quale si comunica che il programma è in forma compressa e pertanto, al fine di ottenere la password necessaria per avviare il processo di decompressione dell'archivio, occorrerà procedere all'invio di alcuni SMS (in genere da uno a tre). Il costo di ciascuno di tali messaggi può addirittura raggiungere i 500 rubli (circa 13 euro)! Una volta effettuato l'invio, l'utente riceverà "in dotazione" un programma maligno, oppure un link ad un sito torrent. In altri casi ancora egli si vedrà recapitare un messaggio di errore o un file archivio... completamente vuoto. La stragrande maggioranza dei computer nei quali è stato individuato il programma malware sopra descritto è situata in paesi russofoni: Russia, Ucraina, Kazakhstan, Bielorussia, Azerbaijan, Moldavia (in ordine decrescente per quantità di computer infettati).

Packed.Win32.Katusha.n

E' un utility di compressione che viene invece impiegata dai malintenzionati per proteggere numerosi programmi malware dall'azione condotta dai software antivirus. Sotto tale denominazione in genere si nascondono dei falsi antivirus, compressi mediante il «packer» Katusha.

Exploit.Java.CVE-2010-0886.a, Exploit.Java.Agent.f e Trojan-Downloader.Java.Agent.jl

Gli ultimi due programmi malware sopra citati sfruttano la vulnerabilità CVE-2010-3867 e vengono caricati sul computer-vittima tramite lo Trojan.JS.Agent.bmh.

Exploit.HTML.CVE-2010-1885.a

E' in pratica costituito da uno script che si avvale della vulnerabilità CVE-2010-1885. Il file che contiene il codice nocivo è costituito da una pagina html, all'interno della quale viene posto un iframe contenente un indirizzo appositamente predisposto dai malintenzionati.



Frammento di Exploit.HTML.CVE-2010-1885.a

Una volta avviato il file, viene effettuato il download di un altro script, identificato come Trojan-Downloader.JS.Psyme.aoy, il quale, a sua volta, provvede in seguito a caricare ed avviare uno dei malware membri della famiglia Trojan-GameThief.Win32.Magania, preposto al furto delle password utilizzate nell'ambito dei giochi online. È interessante rilevare come nello script intermedio venga applicato un metodo piuttosto singolare per occultare il link maligno; quest'ultimo viene difatti scritto «alla rovescia». Ciò risulta evidente nello screenshot qui sotto inserito:



Frammento di Trojan-Downloader.JS.Psyme.aoy: lo script costituisce un anello intermedio all'interno dello schema operativo utilizzato da Exploit.HTML.CVE-2010-1885.a

AdWare.Win32.Boran.z

Questo AdWare era già stato individuato nell'ottobre del 2009: si tratta, in sostanza, di un modulo BHO, distribuito dai malintenzionati assieme al relativo driver di protezione.

Trojan.JS.Agent.bhl

Uno di quei programmi abitualmente utilizzati per diffondere in Rete pubblicità moleste. Si tratta, nella fattispecie, di uno script che provvede ad aprire finestre pop-up indesiderate sugli schermi degli utenti. Per bypassare i sistemi di sicurezza preposti ad impedire l'apertura di tali finestre, esso si avvale di varie tecnologie. Lo screenshot sotto riportato mostra un frammento del file contenente il codice (con relativo commento) elaborato per contrastare l'azione svolta dal modulo popup blocker di Norton Internet Security.

```
//window.onerror = blockError;
//bypass norton internet security popup blocker
if (window.SymRealWinOpen){window.open = SymRealWinOpen;}
if (window.NS_ActualOpen) {window.open = NS_ActualOpen;}
if (typeof(usingClick) == 'undefined') {var usingClick = false
if (typeof(usingObject) == 'undefined') {var usingObject = fa
if (typeof(usingEditor) == 'undefined') {var usingEditor = fa
if (typeof(popwin) == 'undefined') {var popwin = null;}
```

di penelope.di.pixel
redazione@hackerjournal.it



CRACK ME

VIDEOGIOCHI

IN QUESTO ARTICOLO DAL TAGLIO GENERALISTA VOGLIAMO FARE UN PO' IL PUNTO SULLA SICUREZZA DEI SISTEMI DIGITALI DEDICATI AL GIOCO E ALL'INTRATTENIMENTO. C'È CHI GIÀ DA TEMPO HA ALZATO BANDIERA BIANCA, CHI RESISTE E CHI... È INATTACCABILE.

La protezione di un sistema hardware e software destinato all'intrattenimento è una questione dannatamente seria. Un problema che passa attraverso una protezione fisica, insita proprio nell'hardware e una protezione a substrato superiore, software, legata a firmaware e sistema operativo.
Nel corso di questi ultimi anni quasi tutte le console da gioco sono state violate ma ci sono delle eccezioni eccellenti. Discorso un po' diverso per la TV digitale a pagamento che ha vissuto un periodo piuttosto travagliato, per non dire "bucato" per poi diventare pressoché impenetrabile.
In questo breve articolo vedremo un po' qual è lo "stato dell'arte" ovvero quali sono e come si sono evolute le varie modifiche hardware e software applicate ai principali dispositivi di intrattenimento.
Ora la R4 è pronta per ospitare i giochi Nintendo con estensione .nds. Avviando il Nintendo DS dopo averla inserita verrà visualizzata un'interfaccia grafica che mostra l'elenco di tutte le risorse .nds, ovvero dei giochi, disponibili. Basta selezionare un gioco ed avviare, la scheda genererà il file di salvataggio relativo e si potrà iniziare giocare.

NINTENDO DS E DSI

Ne abbiamo parlato un qualche numero fa proprio su Hacker Journal (numero 204) qui il mercato offre tutto quello che serve per fare girare su console giochi scaricati dal web. In particolare per il Nintendo DS tutto quello che serve è l'acquisto della scheda R4 (http://www.r4ds.me/index-en.html) e il download del firmware necessario dal sito del produttore.
Non serve null'altro. Diverso il discorso per il nuovo Dsi. Qui la R4 non funziona bisogna procurarsi una scheda Acekard (HYPERLINK "http://www.acekard.com/" http://www.acekard.com/) che peraltro funziona esattamente come la R4. Una volta in possesso della scheda, dopo aver installato il firmware consigliato dal produttore che serve per installare l'interfaccia grafica e gli elementi preposti al funzionamento, basta semplicemente copiare i giochi sulla scheda (sd inclusa) e, scusate il bisticcio di parole, il "gioco è fatto". I prezzi della R4 variano partendo da 19 euro circa (senza scheda SD inclusa). La Acekard si trova intorno ai 22 euro.

PLAYSTATION 2 E 3

La modifica della Playstation 2, che consente di giocare con titoli masterizzati, è una realtà da anni. Si tratta di una classica modifica che richiede l'installazione di un modchip. Il costo della modifica parte da 59 euro circa.
Dipende dal fornitore, comunque su internet non c'è che l'imbarazzo della scelta. Assai diverso è il discorso della Playstation 3 che, a tutt'oggi, è inviolata. George Hotz ha proposto una modifica del firmware, presentata sul numero 196, che però consente solo di accedere al sistema e non di utilizzare giochi o altro. Si tratta, oltretutto, di una modifica piuttosto empirica il cui esito positivo è tutt'altro che scontato.







Nelle news abbiamo pubblicato la notizia di un dispositivo hardware (chiavetta USB) che consente di mandare in esecuzione su Playstation 3 degli HomeBrew (programmi o giochi). Però è solo una notizia, il dispositivo non è ancora in commercio e quindi la sua reale efficacia è tutta da provare. Metteteci poi che l'eventuale modifica della PS3 arriva alla vigilia del lancio della Playstation 4 e che, quindi, gli hacker dovranno rimettersi al lavoro quasi subito per violare la nuova console, dopo avere impiegato quasi quattro anni per capire come funzionavano i meccanismi di protezione della vecchia, per avere l'esatta idea di cosa sia diventata nel corso degli anni la Playstation: una sorta di fortino inespugnabile.

PSP

Per la playstation portatile il discorso è un po' analogo a quanto detto per la Playstation casalinga.
I primi modelli sono modificabili, si parla quindi delle PSP 1000 e le 2000, ma limitatamente al firmware 3.95.
Le 2000 aventi un firmware dal 4.01 in avanti montano come MotherBoard la TA-088 v3, la quale non permette la modifica, poiché risulta anti-pandora (l'installer usato in abbinata col Despertar del Cementerio).
Le 3000 non si possono modificare, in quanto montano la TA-090, che presenta lo stesso problema della TA-088 v3. Al La TA-090 non permette la modifica software della batteria e non permettono l'installazione di un custom firmware. L'unico modo per poter modificare tali versioni è l'utilizzo dell'exploit ChickHEN sul firmware 5.03. Se tali consolle hanno un firmware superiore, non si possono modificare.
In questo caso si parla di modifiche del tutto simili, come funzionamento, a quelle del Nintendo DS: scheda su cui copiare firmaware e giochi e da inserire nella console.

XBOX 360

Le alternative per la console di casa Microsoft certo non mancano. La tradizionale modifica hardware con modchip si trova un po' ovunque a prezzi intorno ai 45 euro.
Tuttavia è possibile modificare la console anche solo via software, con modifica del firmware e l'ausilio di alcuni strumenti software facilmente reperibili in rete (nei prossimi numeri tratteremo l'argomento da vicino).

WII

Discorso identico a quello già affrontato per XBox. Nel numero 197 abbiamo illustrato, con dovizia di particolari, la modifica software per mandare in esecuzione gli HomeBrew. Con circa 50 euro si può però optare per la modifica hardware con Modchip.

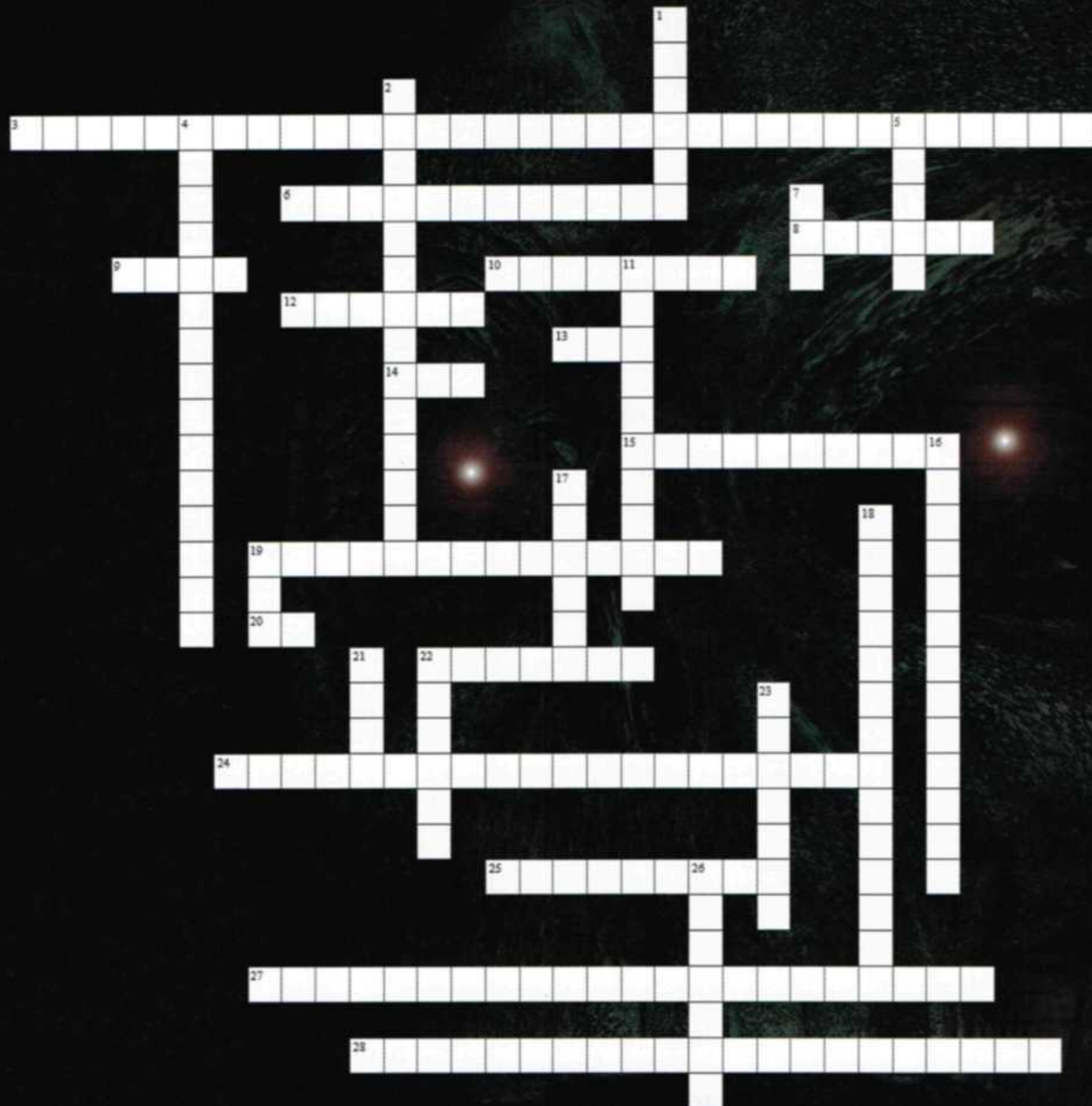
IPHONE 3GS E IPHONE 4

Si parla di jailbreak del sistema operativo 3.0 e successivi e del nuovo 4.0 (montato di serie sui nuovi iPhone 4). Il jailbreak consente di sbloccare il dispositivo e caricare applicazioni senza dovere per forza passare dall'Apple Store. Questo consente sia di caricare applicazioni appositamente sviluppate per circuiti alternativi all'Apple Store, sia di caricare applicazioni originariamente scaricate dall'Apple Store e poi distribuite in rete (praticamente ogni applicazione di successo presente su Apple Store ha un suo omologo non autorizzato circolante in rete, basta cercare. Al di là del puro discorso etico va poi capito se il gioco vale la candela visto che molte app costano appena 79 centesimi...).
Il Jalibreak avviene via software e si può effettuare con diversi pacchetti tra cui: Blackra1n (HYPERLINK "http://www.blackra1n.com" http://www.blackra1n.com) per 3.x o Jailbreakme (HYPERLINK "http://jailbreakme.com/faq.html" http://jailbreakme.com/faq.html). Quest'ultimo valido per l'OS 4.0 e scaricabile direttamente dall'iPhone collegandosi con Safari all'indirizzo sopra riportato.

SKY E DINTORNI

Vi ricordate i primi anni 2000? Bastava tendere un po' l'orecchio per sentire stralci di conversazione dal contenuto piuttosto criptico. Schede da programmare, canali in chiaro da vedere senza spendere una lira, ecc. Era per la precisione il 2004 (vado a memoria) quando cominciarono a girare i primi kit e schede riprogrammabili.
Va detto che questa situazione è durata davvero poco con il passaggio della codifica di segnale da Seca a nds/VideoGuard, sviluppato dalla NDS Group (di origine israeliana ma di proprietà di News Corp, società a cui Sky fa capo), e considerato di gran lunga più resistente alla forzatura.

CYBERENIGMA
HACKER JOURNAL 206



Orizzontali

3. Digest MD5 di Hacker Journal.
6. Anche per il 2010 resta al top delle classifiche come metodologia d'attacco per il web d'eccellenza.
8. Famosa ezine della scena underground.
9. Syscal POSIX-compliant per mappare oggetti (file/device) in memoria.
10. Famoso algoritmo per la ricerca delle Shortest Paths.
12. Ideato ed adoperato da Facebook per incrementare le performance del famoso social network.
13. L'OS utilizzato da Cisco per i suoi device.
14. Copia da a, in asm.
15. Nella realizzazione di software meglio prestare attenzione a non dimenticarsi di de-allocare memoria per evitarli.
19. Una tra le tipologie di vulnerabilità più conosciute. Il tallone d'achille d'eccellenza degli anni 90.
20. Il pesce palla dirige e controlla le reti con lui.
22. Definisce attraverso sinonimi nuove tipologie di dato in C (keyword).
24. Spesso criticata, tra i provdotti ad utilizzarla su larga scala si annoverano i Cisco Service Control Engine.
25. Li si fanno raggiungere attraverso l'esecuzione di un software buggato sovrascrivendo l'indirizzo di ritorno dello stack.
27. Lo sono quelle Liste che mantengono i riferimenti all'elemento successivo e precedente.
28. PHP: andrebbe sempre utilizzata per effettuare l'escape di query a DBMS MySQL (senza underscore).

Verticali

1. Famosa conferenza Hacker.
2. Un uomo nel mezzo spesso troppo curioso.
4. Le stringhe andrebbero opportunamente formattate per evitare bug di questo tipo.
5. Dato strutturato costituito da un insieme finito di valori dello stesso tipo.
7. Su iPhone ce n'è una per tutto.
11. Per l'utente rappresentano il campanello per richiedere l'attenzione del kernel.
16. Secondo Cesare Hacker Journal si scrive così.
17. Identifica la suddivisione di un processo in più parti eseguite pseudo-contemporaneamente dal Calcolatore.
18. A lui si deve l'architettura del Calcolatore generalmente intesa.
19. Protocollo di routing inter-AS d'eccellenza.
21. Quando la voce viaggia su IP.
22. Famoso web container JSP/Servlet.
23. Famosa strategia d'attacco per bypassare i meccanismi di ASLR utilizzati dal kernel Linux a partire dalla release 2.6.
26. Si è fatto conoscere per la sua fama di OS più sicuro al mondo.

LA POSTA DI HJ

SBLOCCO NOKIA

Salve Redazione di HJ, seguo la vostra rivista con grande interesse da ormai molti numeri e un po' mi dispiace per il cambiamento di periodicità, scelta comunque in linea con i cambiamenti che un po' tutti stiamo vivendo in questo periodo. Sono certo che la qualità rimarrà immutata, continuate così! Nell'ultimo numero ho letto un articolo riguardante la cartuccia R4 per Nintendo DS, un argomento di largo intereresse che può abbracciale una fascia di clientela più ampia.
Ho così pensato di suggerirvi di trattare un altro argomento a mio avviso interessante: i "cooked firmware" per gli ultimi smartphone di casa Nokia. Pensiamo ad esempio al popolare 5800 XM: gli ultimi aggiornamenti ufficiali del firmware (v40 e v50) hanno vanificato la procedura di "sblocco" da voi descritta qualche tempo addietro rendendo quindi impossibile svincolarsi dall'ovi store per l'installazione delle applicazioni ed impedendo l'esecuzione di programmi scritti anche puramente a scopo didattico. Un 5800 così conciato ricorda fin troppo il pupillo di casa Apple, non trovate? Sarebbe interessante informare i possessori di questi telefoni delle molteplici possibilità offerte da un firmware modificato, anche solo in fatto di personalizzazione dell'aspetto, dei vantaggi ma anche degli svantaggi introdotti.
Grazie per la vostra attenzione, alla prossima!

In effetti l'argomento è interessante. In questo numero affrontiamo il tema in modo generalista nell'articolo Crack Me, ma è molto probabile che la tua richiesta trovi un riscontro già dai prossimi numeri.

IL CODICE BINARIO MISTERIOSO

Ciao sono Ganryu e mi sono appena iscritto al forum. Non ho seguito la vostra rivista fin dall'inizio quindi forse lo avete gia fatto ma vorrei suggerire di pubblicare articoli sui giochi che stanno attualmente spopolando fra i ragazzi come me, ad esempio MMORPG o browser games. E' solo un'idea che mi è venuta in mente oggi pomeriggio ma mi sentivo di suggerirla. Volevo fare anche una domanda: il codice binario attorno al teschio del logo della rivista ha un significato particolare? L'ho gia provato varie volte in decimale, ottale e esadecimale, nonché anelli uroborici e altre cose ma non ho ancora trovato niente... Mi ha incuriosito e volevo saperne di più.

Saluti, Ganryu

Puoi smetterla di scervellarti perché il codice binario di cui parli ha una funzione meramente decorativa. L'intuizione venne al grafico che ideò quello che da circa otto anni è il logo ufficiale di HJ, ovvero Gianfranco Fagotto che non lavora più con noi ma che ne approfittiamo per salutare con affetto.

ERRATA CORRIGE

Gentilissimi, a seguito del mio articolo "Il Reverse Engineering di FastWeb" pubblicato sul nr° 202 ho erroneamente dimenticato di citare la fonte di tale ricerca. Pertanto vi chiedo, se possibile, di comunicare l'errata corrige in un futuro numero di HJ citando gli autori di tale ricerca: WhiteHatCrew ed il loro blog: http://wifiresearchers.wordpress.com

Andrea Draghetti

Ecco fatto.

PIÙ CHIAREZZA, PLEASE

Cara rivista HJ, io ho iniziato da poco a leggere HJ ma mi è piaciuta subito poiché tratta di argomenti che altre riviste non trattano, purtroppo io so ben poco in fatto di informatica quindi se è possibile volevo darvi un consiglio dal mio punto di vista, perchè quando utilizzate dei termini nuovi alla fine dell'articolo non mettete un minidizionario cosi da far capire meglio articoli complessi a persone poco informate vi ringrazio della vostra disponibilità.

In effetti stiamo pensando ad una sorta di glossario per aiutare i lettori meno esperti a capire meglio alcuni articoli. Tuttavia diversi articoli continueranno a sembrarti di difficile comprensione, proprio perché si tratta di argomenti complessi, che richiedono una serie di conoscenze piuttosto approfondite per poterli apprezzare fino in fondo. E' proprio questa la sfida di HJ. La nostra rivista deve servire da stimolo. Quando un lettore sfoglia una articolo tecnico, magari pieno di codice, se si sente attratto dall'argomento troverà sicuramente lo stimolo per cercare altre fonti altrove (su internet?) che lo aiutino a decifrare l'arcano. Così lettura dopo lettura, quasi inconsapevolmente, il suo patrimonio di conoscenze crescerà fino a consentirgli non solo di leggere gli articoli con estrema disinvoltura, ma di cimentarsi egli stesso nella difficile arte della programmazione. Esageriamo? Non proprio un nostro collaboratore ha iniziato proprio così, per cercare di soddisfare una sua curiosità, e ora programma in C++ senza problemi. La via della conoscenza è irta di ostacoli, ma proprio per questo così affascinante.

Giovanni Federico - info@giovannifederico.net
Fabio `BlackLight` Manganiello - blacklight@autistici.org

PARTE VI

CORSO DI PROGRAMMAZIONE IN C

LINGUAGGI LA SESTA PARTE DEDICATA A STRINGHE E FILE.

Nelle prime parti di questo Corso di Programmazione abbiamo più volte ricordato l'importanza insita nelle definizioni dei dati generalmente intesi così come trattati dal Calcolatore. Abbiamo quindi presentato le metodologie attraverso cui è possibile gestire, tipizzare e lavorare con le informazioni (vedi I Parte - HJ 200).

In questa sesta parte, conclusiva per quanto concerne l'aspetto della "programmazione di base", scopriremo alcuni concetti avanzati del Linguaggio C che consentiranno al neo-programmatore di lavorare su nuovi tipi e capire come gestire più dati omogenei e non attraverso strutture ed unioni.

Analizzeremo poi come, attraverso l'utilizzo delle Liste, sia possibile realizzare strutture dati dinamiche, utilissime per la creazione di software complessi.

Anche in questa sede è opportuno evidenziare che la comprensione di questa trattazione è garantita solo una volta fatte proprie le nozioni sui Puntatori presentate nella quarta parte, disponibile gratuitamente online nella sezione "Archivio" del nostro sito (www.hackerjournal.it) che, ovviamente, ricordiamo essere il punto di riferimento per eccellenza per contattare chi la rivista la scrive e la legge confrontandosi con l'intera community.

STRUTTURE

Il modo più semplice per capire, dal punto di vista pratico, il concetto di Struttura in un qualsiasi linguaggio di programmazione è offrire un confronto con gli Array nell'ipotesi in cui vi sia la necessità di raggruppare un determinato numero di informazioni in uno spazio contiguo e queste non siano necessariamente omogenee. Se infatti volessimo raggruppare un insieme non uniforme di dati, come ad esempio una rubrica telefonica, caratterizzata dall'avere varie etichette per nome, cognome e numero telefonico, il ricorso agli Array non offrirebbe i risultati sperati vista l'eterogeneità delle informazioni da gestire e l'incapacità di gestire dati difformi con gli stessi.
Diamo pertanto la seguente:

> **DEFINIZIONE 27**
> **STRUTTURA**
> Si definisce "struttura" un qualsiasi insieme logico di variabili non necessariamente caratterizzate dall'appartenenza allo stesso tipo. Una struttura può costituire, a sua volta, un tipo di variabile previsto all'interno di un'ulteriore struttura. Parliamo in questo caso di Struttura di Strutture.

La dichiarazione di una struttura presuppone l'utilizzo di una nuova keyword del linguaggio, struct, alla quale si fa seguire il nome arbitrario scelto per la stessa e, delimitato da parentesi graffe, l'elenco delle variabili che dovrà contenere nella classica forma tipo nome_variabile.

```
struct Nome_Struttura {
        tipo Nome_Variabile1,
        tipo Nome_Variabile2,
        ...
        tipo Nome_VariabileN;
};
```

È decisamente fondamentale ricordare che la dichiarazione di una struttura, di fatto, definisce un nuovo tipo di dato, dal momento della creazione in poi disponibile insieme ai tipi fondamentali del linguaggio visti nelle scorse parti del Corso.

Come abbiamo citato nella definizione 27, una Struttura può essere a sua volta elemento costituente di un'altra Struttura, in questo caso la dichiarazione non cambia se non per il fatto che come tipo di variabili indicate all'atto della dichiarazione intenderemo strutture:

```
struct Gruppo_Strutture {
        struct Nome_Struttura1;
        struct Nome_Struttura2;
        struct Nome_StrutturaN;
};
```

INIZIALIZZAZIONE ED ACCESSO A STRUTTURE

L'utilizzo di nuovi tipi di dati definiti dal programmatore attraverso il sapiente utilizzo di Strutture non si esaurisce naturalmente all'atto della dichiarazione della Struttura stessa. Prendiamo in considerazione la seguente Struttura, occupandoci immediatamente dopo averla dichiarata di inizializzarla attraverso la dichiarazione di una variabile del tipo appena definito:

```
struct HackerJournal {
        int numero;
        char data[8];
};
```

Pertanto, per dichiarare una variabile di tipo "HackerJournal" da ora scriveremo:

```
struct HackerJournal nome_variabile;
```

In modo del tutto equivalente la dichiarazione della Struttura "HackerJournal" e della variabile di tipo "HackerJournal" appena vista può essere sintetizzata in un unico blocco di codice nella forma:

```
struct HackerJournal {
        int numero;
        char data[8];
} nome_variabile;
```

L'accesso ai campi di una struttura avviene per mezzo dell'operatore "punto" ("."), posto in modo da individuare il nome della variabile di tipo struttura sulla sinistra ed il nome dell'elemento al quale si vuole accedere a destra.
Offriamo, dunque, un primissimo esempio di utilizzo di una Struttura dati attraverso il seguente codice:

```
#include <stdio.h>

struct HackerJournal {
   int numero;
   char data[8];
};

int main()
{
   struct HackerJournal in_edicola;

   in_edicola.numero = 206;
   strcpy(in_edicola.data,
"23092010");

   printf("Il numero %d della rivista
e' in edicola dal %s", in_edicola.
numero, in_edicola.data);
   return 0;
}
```

Compilando ed osservando l'output dovrebbe apparire a questo punto chiaro quanto appena realizzato:

```
$ gcc hjstruct.c -o hjstruct
$ ./hjstruct
Il numero 206 della rivista e' in edi-
cola dal 23092010
```

Ricordiamo, anche in questo caso (e fino alla noia), di prestare sempre particolare attenzione a scritture di questo tipo: strcpy(in_edicola.data, "23092010");. I motivi dovrebbero essere decisamente chiari al lettore a questo punto del Corso.

Come nel caso della dichiarazione, anche quanto visto nel primo esempio offerto poteva sinteticamente venire espresso ricorrendo ad un unico blocco di istruzioni, nella forma:

```
struct HackerJournal {
      int numero;
      char data[8];
} in_edicola = { 206, "23092010" }
```

STRUTTURE E FUNZIONI

Le Strutture, come qualsiasi tipologia di dato in C, possono essere parametri o valori di ritorno di funzioni.

Nel caso in cui siano parametri di funzioni queste potranno, ovviamente, essere passate per valore o per riferimento. Nel primo caso, come sappiamo, i dati verranno copiati in una zona distinta di memoria accessibile nell'ambito locale della funzione: le modifiche, pertanto, non influenzeranno la struttura originaria.

Nel secondo, utilizzando come immaginabile un puntatore a struttura, ci riferiremo all'esatta locazione di memoria della Struttura stessa, rendendo questa soggetta a modifiche da parte della funzione considerata.

Consideriamo il seguente sorgente esemplificativo che ci servirà, peraltro, ad introdurre una nuova importantissima keyword del linguaggio.

```
#include <stdio.h>

struct HackerJournal {
      int numero;
      char data[8];
};

typedef struct HackerJournal HJ;
void Riempimento(HJ *ptr);

int main()
{
   HJ in_edicola;
   Riempimento(&in_edicola);
   printf("Il numero %d della rivista
e' in edicola dal: %s\n", in_edicola.
```

```
numero, in_edicola.data);
   return 0;
}

void Riempimento(HJ *ptr){
   printf("Inserire il numero della
rivista: ");
   scanf("%d", &ptr->numero);
   printf("Inserire data della rivista in
formato GGMMAAAA: ");
   scanf("%s", ptr->data);
}
```

Prima di commentare il codice scoprendo alcune novità presentate, compiliamolo ed osserviamo l'output:

```
$ gcc hjstruct2.c -o hjstruct2
$ ./ hjstruct2
```

Inserire il numero della rivista: 205
Inserire data della rivista in formato GGMMAAAA: 23092010
Il numero 205 della rivista e' in edicola dal: 23092010

L'utilizzo della keyword typedef combinata alla Struttura creata ci ha permesso di soprannominare "HJ" il tipo Struttura HackerJournal, rendendo più compatto e leggibile il codice: non abbiamo pertanto più dichiarato la variabile "in_edicola" come "struct HackerJournal in_edicola;" ma, più semplicemente, "HJ in_edicola;".
Su questa abbiamo poi effettuato tutte le valorizzazioni spendibili sul tipo Struttura.

La funzione Riempimento(), il cui funzionamento dovrebbe essere praticamente chiaro, riceve come parametro il puntatore alla Struttura realizzata (si tratta quindi di un passaggio per riferimento) e su questa effettua le operazioni di assegnazione utilizzando stavolta l'operatore "->" (da utilizzare ogniqualvolta si adoperano puntatori a strutture).

Infine, nel corpo della main(), stampiamo a video i campi "numero" e "data" della Struttura.

In modo più compatto, inoltre, avremmo potuto definire il nuovo tipo HJ combinando l'uso di typedef e struct in un unico blocco, come di seguito:

```
typedef struct HackerJournal {
       int numero;
       char data[8];
} HJ;
```

Giunge pertanto il momento di offrire la seguente:

DEFINIZIONE 28
TYPEDEF

La keyword typedef consente l'assegnazione di un nome arbitrario ad un tipo, permettendo, di fatto, la realizzazione e l'utilizzo di nuovi tipi in modo compatto e leggibile.

Rimanendo in tema di Strutture, offriamo un semplice esempio di come, attraverso l'uso della typedef, sia possibile identificare due nuovi tipi nei nostri sorgenti identificati a mezzo di un'etichetta scelta dal programmatore: uno riferito alla struttura HackerJournal vista prima ed uno riferito al puntatore alla struttura stessa:

```
typedef struct HackerJournal HJ;
typedef struct HackerJournal *
ptrHJ;
```

Tramite questa scrittura potremo riferirci alla Struttura HackerJournal utilizzando semplicemente il tipo "HJ" ed al puntatore alla Struttura utilizzando "ptrHJ" in qualsiasi parte del sorgente.

È inoltre decisamente poco opinionabile che dichiarare una variabile di tipo Struttura HackerJournal risulti sicuramente più comodo nella forma "HJ nome_variabile;" piuttosto che "struct HackerJournal nome_variabile;", così come riferirsi ad un puntatore alla stessa risulti decisamente più immediato con "ptrHJ puntatore;" piuttosto che "struct HackerJournal * puntatore".

Riferendoci alle scorse parti del Corso sugli Array, inoltre, potrebbe risultare più elegante definire ed utilizzare un tipo Matrice come:

```
typedef Matrice int[MAX_RIG][MAX_
RIG];
```

NOTA IMPORTANTE

Il meccanismo della typedef potrebbe essere erroneamente considerato identico a quello della #define. Per quanto queste due keyword lavorino in modo simile realizzando una sostituzione del nome con tutto quanto indicato dopo il nome stesso, la typedef lavora esclusivamente sulla dichiarazione di tipi a differenza della #define che applica sostituzioni di qualsiasi natura in ogni parte del sorgente.

Concludendo, come qualcuno ricorderà, abbiamo già incontrato funzioni e strutture nella scorsa parte, più precisamente quando abbiamo lavorato con i FILE e più volte ci siamo riferiti alla struttura FILE della libreria Standard Input/Output (stdio.h), la cui dichiarazione, oramai comprensibile al lettore, varia da Sistema Operativo a Sistema Operativo ma risulta contenere sempre campi indicanti informazioni sul file quali nome, dimensione, descrittore, flag, modalità di accesso, buffer, eccetera.

Per intenderci, una tipica struttura FILE potrebbe essere la seguente (avendo eliminato qualsiasi commento, sfidiamo il lettore a capire a che OS ci siamo riferiti):

```
typedef struct __sFILE {
       unsigned char *_p;
       int      _r;
       int      _w;
       short    _flags;
       short    _file;
       struct   __sbuf _bf;
       int      _lbfsize;
       void     *_cookie;
       int      (*_close)(void *);
       int      (*_read)(void *, char
```

```
*, int);
        fpos_t    (*_seek)(void *,
fpos_t, int);
        int       (*_write)(void *,
const char *, int);
        struct    __sbuf _ext;
        unsigned char *_up;
        int       _ur;
        unsigned char _ubuf[3];
        unsigned char _nbuf[1];
        struct    __sbuf _lb;
        int       _blksize;
        fpos_t    _offset;
} FILE;
```

Ci aspettiamo che ora siano maggiormente chiari da parte del lettore i concetti appresi nella quinta parte del Corso.

STRUTTURE DI STRUTTURE

Abbiamo già detto che campi di una Struttura possono essere Strutture stesse. L'esempio più semplice al quale possiamo riferirci per illustrare l'uso di Strutture di Strutture è quello che tutti voi conosceranno del rappresentare una retta nel piano cartesiano.
Creeremo dapprima un nuovo tipo di dato per individuare il punto sul piano (Point) e che conservi quindi le coordinate sull'asse X ed Y. Adopereremo quindi un ulteriore tipo ad hoc per la retta (Line) costituito da due punti (due Point).

```
#include <stdio.h>

typedef struct {
   int x;
   int y;
} Point;

typedef struct {
   Point start;
   Point end;
} Line;

int main()
{
   // Dichiariamo 2 punti, per i quali
abbiamo creato il tipo apposito:
   Point a, b;
   a.x = 10;
   a.y = 5;
   b.x = 20;
   b.y = 10;

   // Una retta passa sempre per due
punti, anche per questa abbiamo
creato il tipo apposito
   Line retta;
   retta.start = a;
   retta.end = b;

   printf("Primo punto: %d(asse X)
%d(asse Y)\n", retta.start);
   printf("Secondo punto: %d(asse
X) %d(asse Y)\n", retta.end);
}
```

Compiliamo ed osserviamo se l'output è esatto:

```
$ gcc line.c -o line
$ ./line
Primo punto: 10(asse X) 5(asse Y)
Secondo punto: 20(asse X) 10(asse
Y)
```

UNIONI

Concetto simile per modalità di scrittura in codice alle Strutture ma totalmente dissimile per funzionamento e gestione della memoria del Calcolatore sono le unioni, per le quali offriamo immediatamente la seguente:

> **DEFINIZIONE 29**
> **UNIONI**
> Le unioni rappresentano un insieme di variabili di tipo eterogeneo che condividono la stessa locazione di memoria ed utilizzabili in modo esclusivo.

La dichiarazione è del tutto simile a quanto visto per le Strutture ed è la seguente:

```
union Nome_Unione {
        tipo nome_variabile1;
        tipo nome_variabile2;
        ...
        tipo nome_variabile3;
};
```

Per rendere maggiormente chiaro il concetto di unione, possiamo immaginarle (avendo bene a mente che quanto di seguito non è assolutamente una definizione formale corretta) come variabili di tipo multiplo che possono essere adoperate secondo il tipo richiesto al momento dell'utilizzo.

Fondamentale tenere a mente che l'inizializzazione secondo un tipo è univoca: avendo un'unione che può essere usata, ad esempio, come intero o carattere, al momento dell'utilizzo della stessa come un tipo distruggeremo il valore dell'altro tipo.
Compilare ed eseguire il seguente sorgente renderà maggiormente chiare le idee:

```
#include <stdio.h>

int main()
{
   union {
      int a;
      char b;
   } unione;

   unione.a = 65;
   fprintf(stdout, "%d - %c\n",
unione.a, unione.b);

   unione.b = 'A';
   fprintf(stdout, "%d - %c\n",
unione.a, unione.b);

   return 0;
}

$ gcc unione.c -o unione
$ ./unione
65 - A
65 - A
```

L'output in ambo i casi non cambia, condividendo la stessa locazione di memoria, attraverso l'unione in un caso (unione.a) ci siamo riferiti all'intero 65 mostrato come carattere "A" nella prima fprintf() (unione.b), nell'altro (unione.b) indicavamo il carattere "A" che, come ci aspet-

tavamo, viene ripreso dalla fprintf() come "unione.a" da intero.

In altre parole, al momento dell'inizializzazione di "unione.a" come 65, "unione.b", condividendo la stessa locazione di memoria, è automaticamente inteso come il carattere "A" e viceversa.

L'utilità di questa modalità di trattare il dato è facilmente riscontrabile ogni volta che vi sia la necessità di lavorare con variabili che, durante l'esecuzione del software, possono assumere (in momenti distinti) tipi diversi.

ENUMERAZIONI

Come il nome stesso suggerisce, il Linguaggio C consente al programmatore di "enumerare" un insieme di dati attraverso l'utilizzo del costrutto enumerazione (enum). Questo insieme è quindi assunto nel software come un tipo di dato a sé stante, ordinato ed arbitrariamente definito in fase di progettazione dell'applicativo.

Iniziamo col dare una immediata:

DEFINIZIONE 30

TIPO ENUMERATIVO

Il tipo per enumerazione "enum" consente di associare valori interi a costanti letterali (enumeratori).

Scopriamo un semplice esempio per capire come potrebbe essere utilizzata l'enumerazione:

```
#include <stdio.h>

typedef enum {
   Lunedi = 1,
   Martedi,
   Mercoledi,
   Giovedi,
   Venerdi,
   Sabato,
   Domenica
} giorno;

int main()
{
   giorno i;
   for(i = Lunedi; i <= Domenica; i++)
      printf("Operazione eseguita dal lun alla dom. (%d giorno della settimana)", i);
}

$ gcc enum.c -o enum
$ ./enum
```

Operazione eseguita dal lun alla dom. (1 giorno della settimana)

Operazione eseguita dal lun alla dom. (2 giorno della settimana)

Operazione eseguita dal lun alla dom. (3 giorno della settimana)

Operazione eseguita dal lun alla dom. (4 giorno della settimana)

Operazione eseguita dal lun alla dom. (5 giorno della settimana)

Operazione eseguita dal lun alla dom. (6 giorno della settimana)

Operazione eseguita dal lun alla dom. (7 giorno della settimana)

Questo sorgente, per quanto sia utile solo a fini didattici, evidenzia come sia possibile realizzare un nuovo tipo di dato (nel nostro caso i giorni della settimana) e lavorare con questo come per qualsiasi tipologia di dato fondamentale del Linguaggio C.

Per capire ciò basta osservare com'è inizializzato il ciclo for (for(i = Lunedi; i <= Domenica; i++)), allo stesso modo di quanto si farebbe per gli interi, in questo caso lavoriamo sul tipo "giorno" da noi definito.

Importante notare, inoltre, che una volta definito il primo enumeratore, i successivi saranno autonomamente incrementati a partire dal quel numero (nel nostro caso 1).
Come prevedibile, qualora non fornissimo alcun indice di partenza, l'enumerazione partirebbe autonomamente da 0.

Ma vediamo qualcosa di più utile. Come abbiamo avuto modo di apprendere nel corso di queste trattazioni, il Linguaggio C non dispone, tra i tipi predefiniti, di quello booleano. Una semplicissima scrittura fornirà ai nostri sorgenti ciò che l'ANSI C non "incorpora" di default ed ovvero:

```
typedef enum { false, true } Booleano;
```

LISTE

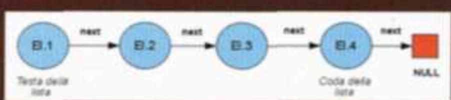
Una volta padroneggiato l'insieme di meccanismi per creare in C nuovi tipi di dato (typedef, strutture, unioni ed enumerazioni) possiamo usare queste conoscenze per esprimere nuove strutture dati che ci consentano di gestire informazioni in modo più dinamico e snello dei classici array visti finora.

Gli array sono infatti semplici da gestire, ma sono scarsamente dinamici (è possibile allocarli dinamicamente e modificare la loro dimensione, come abbiamo visto, usando la realloc in caso di necessità, ma quest'operazione è molto dispendiosa dal punto di vista computazionale operando su un'area di memoria complessa e tendenzialmente più lenta come lo heap, e non andrebbe abusata), e soprattutto è molto complesso l'inserimento o la cancellazione di un elemento nel mezzo (nel caso dell'inserimento andrebbe allocato un nuovo elemento sull'array, andrebbero spostati in avanti di una posizione tutti gli elementi dall'indice dell'inserimento in poi, e inserito il nuovo elemento nella nuova locazione che si è venuta a creare, mentre invece nel caso di cancellazione bisogna spostare indietro di una posizione tutti gli elementi dall'indice di inserimento in poi e deallocare l'ultimo elemento, che è ora vuoto).

L'array è quindi una struttura dati semplice da gestire e anche veloce nel caso in cui non vengono effettuati molti inserimenti o cancellazioni, ma poco dinamica.

Se vogliamo una collezione di dati meglio gestibile dinamicamente possiamo definirla come lista, dando immediatamente la seguente:

DEFINIZIONE 31
LISTE
Una lista è un insieme di elementi (nodi) collegati l'uno all'altro semplicemente dal riferimento all'elemento successore (si parla in questo caso di lista concatenata). Se il successore di un elemento è NULL, allora quell'elemento è l'ultimo della lista, come illustrato in figura 1.



Lista con elemento NULL come successore.

Da un punto di vista di codice può essere rappresentato come qualcosa del genere:

```
struct node {
        int value;
        struct node* next;
};

/* Tipo predefinito tenuto per comodità */
typedef struct node* list;
```

Esamineremo per ora le liste di tipo LIFO (Last In, First Out), o stack, che sono concettualmente le più semplici da gestire dal punto di vista della programmazione. In questo tipo di strutture dati si può distinguere la testa (head) della lista, che è l'elemento in cima, dalla coda (tail), che nel caso dello stack è il puntatore al prossimo elemento della testa (quindi tutto ciò che c'è dopo la testa è identificato come coda).
Ogni volta che vorremo inserire un nuovo elemento lo inseriremo in testa, dicendo "questa è la nuova testa, e tutto ciò che c'era prima nella lista sarà la sua coda".
Un po' come una pila di dischi infilata in un paletto di legno (si pensi alla famosa torre di Hanoi), in cui ogni volta che vogliamo inserire un nuovo disco lo facciamo mettendolo in testa.
Tale lista può essere vista come un oggetto contenente una sequenza ordinata di valori e su cui è possibile operare diverse azioni (mantenendo come codice del tipo la struct appena illustrata). Fra queste:

INSERIMENTO DI UN NUOVO NODO IN CIMA ALLA LISTA (PUSH)

Metodo che alloca la memoria per un nuovo nodo (singolo elemento), settando come suo successore una lista e diventando quindi la sua nuova "testa". Si noti il passaggio di list come puntatore. list, come abbiamo già visto prima, è già definito come un puntatore (typedef struct node* list). Viene quindi passato come puntatore a puntatore, passaggio necessario per fare in modo che sia modificato anche al di fuori della funzione:

```
void push (int value, list* tail) {
        list l = (list)
malloc(sizeof(struct node));
        l->value = value;
        l->next = *tail;
        *tail = l;
}
```

RITORNO DELL'ELEMENTO DI CIMA DELLA LISTA

```
list head ( list l ) {
        return l;
}
```

RITORNO DELLA CODA DELLA LISTA

```
list tail ( list l ) {
        return l->next;
}
```

LETTURA DELL'ELEMENTO DI TESTA DELLA LISTA E RIMOZIONE DI QUESTO DALLA LISTA STESSA (POP)

```
list pop ( list* l ) {
        list h = head(*l);
        *l = tail(*l);
        return h;
}
```

CONTROLLO SE LA LISTA È VUOTA O MENO

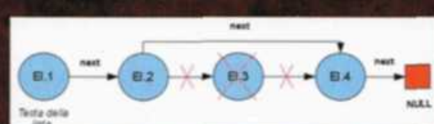
Per fare ciò, dato che a differenza del C++ il C non supporta nativamente il tipo booleano (vero o falso), una soluzione pulita può essere quella di definire "artificialmente" il tipo booleano attraverso una enum come appreso nell'apposito paragrafo:

```
typedef enum { false, true } BOOL;

BOOL empty ( list l ) {
        return ( l == NULL );
}
```

CANCELLAZIONE DI UN ELEMENTO CON UN CERTO VALORE DALLA LISTA

Ciò può essere implementato attraverso un algoritmo ricorsivo.
L'algoritmo di cancellazione può apparire relativamente complesso ma è concettualmente abbastanza semplice. Si tratta di "scansionare" la lista elemento per elemento, e controllare se il successore dell'elemento corrente è quello da rimuovere. Se lo è, allora rimuoveremo l'arco, il collegamento che collega l'elemento corrente a quell'elemento, diremo al programma che il nuovo successore dell'elemento corrente è quello che prima era il successore dell'elemento da cancellare e rimuoveremo definitivamente l'elemento.
L'algoritmo è sinteticamente illustrato in figura 2, dove "El.3" è l'elemento da cancellare.



AR: cancellazione elemento da Lista.

Vediamo pertanto il codice:

```
void delete ( int value, list* l ) {
        list node, next;
```

```
        // Se la lista è vuota, termi-
no
        if ( empty(*l) )
                return;

        // Se il successore è vuoto,
termino
        if ( empty((*l)->next) )
                return;

        node = (*l)->next;

        // Se il successore contiene
l'elemento che cerco
        if ( node->value == value ) {
                next = node->next;
                (*l)->next = next;

                free(node);
                return;
        }

        // Richiama l'algoritmo
ricorsivamente
        delete ( val, &node );
}
```

DEALLOCAZIONE DELLA LISTA

La lista è una struttura dinamica allocata sullo heap via malloc, ed è quindi importante deallocarla quando non serve più.

Tuttavia richiamando la free sulla lista non si fa altro che deallocare il nodo di testa, mentre tutti i nodi successivi rimangono intatti. Anche per la deallocazione l'algoritmo più efficiente è quindi di tipo ricorsivo.

Parte dalla testa della lista, esce se l'elemento corrente è vuoto, mentre invece se il suo successore è vuoto dealloca l'elemento corrente (vuol dire che siamo arrivato alla fine della lista, quindi possiamo cominciare a deallocare andando a ritroso) e ritorna. Altrimenti, richiamiamo l'algoritmo ricorsivamente sull'elemento successivo.

```
void free_list ( list l ) {
  if ( empty(l) )
    return;

  if ( empty(tail(l)) ) {
    free(l);
    return;
  }

  free_list ( tail(l) );
  free(l);
}
```

RICERCA DI UN VALORE

La ricerca di un valore in una lista è estremamente semplice usando un algoritmo ricorsivo, che ritorna NULL se il valore non è trovato o il puntatore al nodo contenente quel valore in caso contrario.

Semplicemente, ritorna NULL se la lista è vuota, ritorna l'elemento corrente se contiene il valore da cercare, altrimenti richiama ricorsivamente l'algoritmo sull'elemento successivo.

```
list search ( int value, list l ) {
  if ( empty(l) )
    return NULL;

  if ( l->value == value )
    return l;

  return search ( value, tail(l) );
}
```

STAMPA DEGLI ELEMENTI DI UNA LISTA

Può essere comodo tenere una funzione che stampi il contenuto di una lista, ricordando che in questa funzione useremo direttamente la lista passata come parametro per "esplorare" tutti gli elementi, in quanto tale argomento viene passato per valore e non per riferimento (quindi le modifiche vengono eseguite solo sullo stack della funzione).

Ma in altri casi, l'esplorazione su tutta la lista è un'operazione distruttiva, almeno nelle liste che non sono doppiamente concatenate (ovvero dove c'è solo il riferimento all'elemento successivo e non al precedente), in quanto assegnando di volta alla lista il nodo successivo si perde il riferimento di quello che c'era prima.

Il primo problema è, come evidente, il non poter più recuperare la lista originale dopo un'operazione del genere. Il secondo, meno evidente ma non meno pericoloso, sta nel fatto che perdendo i riferimenti dei puntatori precedenti non potremo più operare una corretta deallocazione della lista quando non ci serve più, e questo porta a memory leak.

Quindi ricordiamo che in questo caso va bene perché stiamo operando su una copia della lista passata a una funzione, in altri casi è bene, prima di partire per una "esplorazione" degli elementi della lista, fare una copia della lista originale (ad esempio list tmp = l) e operare su quella invece che sulla lista originale.

```
void print ( list l ) {
        while ( !empty(l) ) {
                list el = pop( &l );
                printf ("%d -> ", el-
>value);
        }

        printf ("\n");
}
```

ALL'OPERA

Vediamo ora un esempio d'uso delle funzioni viste finora per la gestione di una lista elementare:

```
int main () {
  list l = NULL;

  // Salva 4 elementi sulla lista
  // N.B. la lista è LIFO, gli elementi
  // verranno salvati in ordine inverso
  // quindi la lista finale sarà
  // 4 -> 3 -> 2 -> 1
  push ( 1, &l );
  push ( 2, &l );
  push ( 3, &l );
  push ( 4, &l );
```

```
  // Stampa la lista
  print ( l );

  // Cancella dalla lista l'elemento
con valore 2
  delete( 2, &l );

  // Stampa la lista
  print ( l );

  if ( search(2, l) )
    printf ("L'elemento di valore 1 è
stato trovato\n");
  else
    printf ("L'elemento di valore 1 non
è stato trovato\n");

  // Dealloca la lista
  free_list(l);
  return 0;
}
```

La lista semplicemente concatenata di tipo LIFO che abbiamo esaminato finora è il tipo di struttura dati dinamica più elementare. Ce ne sono di più complesse, dalla lista doppiamente concatenata (ovvero dove ogni nodo tiene non solo il riferimento all'elemento successivo, ma anche a quello precedente), all'albero binario, al grafo completo. Con le conoscenze acquisite finora potete già scrivere un'elementare implementazione di una lista doppiamente concatenata.

In seguito vedremo strutture dinamiche più complesse, quali alberi binari, n-ari e grafi completi (o ciclici). Man mano che la complessità della struttura dinamica aumenta anche la complessità degli algoritmi per la sua gestione, il rischio di errori nel codice, e i tempi di CPU richiesti per gestire correttamente la struttura.

CONCLUSIONI

Volge al termine con questa trattazione il Corso di Programmazione in senso stretto.

Con questo intendiamo dire che i concetti finora illustrati sono quelli da conoscere per poter sviluppare software anche di discreta complessità: rappresentano quindi il bagaglio di conoscenze minime che qualsiasi programmatore deve possedere per definirsi tale.

Nel corso delle prossime tre parti, infatti, saranno presentati argomenti la cui finalità non è didattica, bensì è tesa ad introdurre determinate nozioni relative a tecniche di programmazione raffinate per l'ingegnerizzazione e lo sviluppo del Software.

Parleremo pertanto, se pur in modo introduttivo dovendoci scontrare con oggettivi limiti di stampa, di programmazione multithreading e di rete; impareremo quindi a conoscere ed utilizzare strutture dati complesse, alberi e grafi ed affronteremo una tematica tanto avanzata quanto affascinante: l'intelligenza artificiale e le possibili implementazioni in codice di ogni giorno (vi siete mai chiesti come funziona un navigatore satellitare o come sarebbe possibile realizzare un sistema di sicurezza proattivo "intelligente", magari un IDS neurale ?).

Infine, chiuderemo definitivamente il Corso offrendo una panoramica della programmazione orientata agli oggetti in C, scoprendo, nell'ultima tranche, un nuovo linguaggio di programmazione: il C++.
Non ci resta che dirvi... continuate a seguirci!

UN PRIMO ESERCIZIO IN ATTESA DEL TEST

In attesa del Test proposto al termine del Corso, avendo concluso, come detto prima, la parte de "i fondamenti della programmazione in C" vogliamo chiudere questa trattazione offrendo ai lettori un semplice esercizio con il quale esercitarsi facendo definitivamente proprie tutte le nozioni finora apprese nel corso di queste sei parti.

Non c'è bisogno di dire che non si tratta di una sfida tra utenti, ma semplicemente di un modo per far pratica su cose finora affrontate esclusivamente teoricamente: barare copiando codice preso online non serve dunque a nulla.

L'esercizio proposto è, pertanto, la realizzazione di una rubrica che conservi nome, cognome, indirizzo, recapiti telefonici, email e note all'interno di un database proprietario con estensione ".rub" (da creare o modificare solo al termine del programma) e che consenta, come ci si aspetterebbe da un applicativo del genere, di inserire, modificare, ricercare, eliminare ed esportare in formato ".txt" i contatti.

Il posto dove eventualmente confrontarsi, manco a dirlo, resta il forum dedicato del nostro sito, raggiungibile all'indirizzo www.hackerjournal.it/corsoc.htm. Buon divertimento!